JHipster Side-by-side Blueprint for Multiple Human-Readable Foreign Key Attributes in Angular Client’s User Interface

In this article, the word attribute is equivalent to the word field as they relate to entities. Both the words field and attribute will be used interchangeably.

This is a side-by-side JHipster 7 blue print showing how to put human-readable foreign key field in place of UUIDs into the user interface where entities have relationships with other entities. Sometimes having a UUID makes it difficult for the human in the loop to figure out what the entity on the other side actually is (without clicking on the UUID link). This solution makes it easier for the user to see what the other related entity actually is. Futhermore, this solution is based on adding @customAnnotation’s to entity field(s), in the JDL file, that would be shown in replacement of UUIDs. If multiple entity fields are necessary to replace the UUID, the fields can be delimmited via a specified delimiter using a @customAnnotation, as well.

There is a solution in the JDL syntax to display a field is documented here: https://www.jhipster.tech/jdl/relationships. The JDL syntax to display a field is:

```
relationship (OneToMany | ManyToOne | OneToOne | ManyToMany) {
  <from entity>[{<relationship name>[(<display field>)]}] to <to
entity>[{<relationship name>[(<display field>)]}]+
}
```

An example of the above built-in syntax is (from Matt Raible’s "Micro Frontends for Java Microservices" article / github.com code is below):

```
entity Blog {
  name String required minlength(3)
  handle String required minlength(2)
}

entity Post {
  title String required
  content TextBlob required
  date Instant required
}

relationship ManyToOne {
  Blog{user(login)} to User
  Post{blog(name)} to Blog
}
```

In the above example, the blog's UUID will be replaced by the blog's name in the user interface wher the post will be displayed.

However, the built-in JDL syntax does not provide the ability to display multiple fields to futher distinguish a complex entity. For example, if multiple blogs can have the same name, we would not be able to tell which blog is in the relationship with the post. If we could dispaly the blog's name and handle (assuming no 2 blogs can have the same handle), then we would be able to tell which blog is related to the post. Consequently, this blueprint serves the purpose of relpacing UUIDs with multiple attributes/fields of the other-side relationship with a given entity to help differentiate the entity's relationship on the user interface.

This example also show how to create a JHipster 7 blueprint.

Prerequisites:

- Java 11+
- Node.js 16+
- Docker Desktop
- JHipster 7.9.3

Table of Contents

Setup for Generating JHipster Blueprints 2
Build JHipster Blueprint for Multiple Human-readable Foreign Key Fields. 4
    Use the Blueprint Locally to Generate Code 7
    Modify the Blueprint to Enable Multiple Human-readable Foreign key Fields. 7
Have Fun with Creating JHipster Blueprints! 7

Setup for Generating JHipster Blueprints

1. Install JHipster 7.9.3:

   ```
   npm i -g generator-jhipster@7.9.3
   ```

2. There is a bug, at least on macOS Ventura 13.1, that requires a slight change to the yeoman-environment's package.json file. You will see an error when running the command:

   ```
   jhipster --version
   ```

3. If you see the following error, there is an easy fix below:

```
INFO! Using bundled JHipster
node:internal/modules/cjs/loader:556
      throw e;
      ^

Error [ERR_PACKAGE_PATH_NOT_EXPORTED]: Package subpath './lib/util/namespace' is
not defined by "exports" in /usr/local/lib/node_modules/generator-
jhipster/node_modules/yeoman-environment/package.json
    at new NodeError (node:internal/errors:399:5)
    at exportsNotFound (node:internal/modules/esm/resolve:266:10)
    at packageExportsResolve (node:internal/modules/esm/resolve:602:9)
    at resolveExports (node:internal/modules/cjs/loader:550:36)
    at Module._findPath (node:internal/modules/cjs/loader:619:31)
    at Module._resolveFilename (node:internal/modules/cjs/loader:1046:27)
    at Module._load (node:internal/modules/cjs/loader:905:27)
    at Module.require (node:internal/modules/cjs/loader:1127:19)
    at require (node:internal/modules/helpers:112:18)
    at Object.<anonymous> (/usr/local/lib/node_modules/generator-
jhipster/utils/blueprint.js:19:25) {
  code: 'ERR_PACKAGE_PATH_NOT_EXPORTED'
}

Node.js v19.6.0
```

4. Add the line "./lib/util/namespace": "./lib/util/namespace.js" to the global yoeman-environment's package.json at the end of the "exports" section as follows. You will need to add a "," after "./package.json".

```
vi /usr/local/lib/node_modules/generator-jhipster/node_modules/yeoman-
environment/package.json
```

```
  "exports": {
    ".": "./lib/environment.js",
    "./cli/": "./cli/",
    "./lib/": "./lib/",
    "./lib/util/": "./lib/util/",
    "./adapter": "./lib/adapter.js",
    "./conflicter": "./lib/util/conflicter.js",
    "./log": "./lib/util/log.js",
    "./transform": "./lib/util/transform.js",
    "./package.json": "./package.json",
    "./lib/util/namespace": "./lib/util/namespace.js"
  },
```

5. Now run the jhipster version command again and you will see the version as below.

```
jhipster --version
INFO! Using bundled JHipster
7.9.3
```

Build JHipster Blueprint for Multiple Human-readable Foreign Key Fields.

1. To generate a JHipster blueprint, run the following command:

```
cd ~
mkdir generator-jhipster-multiple-human-readable-foreign-key-fields
cd generator-jhipster-multiple-human-readable-foreign-key-fields
jhipster generate-blueprint

INFO! Using bundled JHipster

        ██╗ ██╗   ██╗ ████████╗ ███████╗   ██████╗ ████████╗ ████████╗ ███████╗
        ██║ ██║   ██║ ╚══██╔══╝ ██╔═══██╗ ██╔════╝
╚══██╔══╝ ██╔═════╝ ██╔═══██╗
        ██║ ████████║    ██║    ███████╔╝ ╚█████╗     ██║    ██████╗   ███████╔╝
  ██╗   ██║ ██╔═══██║    ██║    ██╔════╝   ╚═══██╗    ██║
██╔═════╝   ██╔═══██║
  ╚██████╔╝ ██║   ██║ ████████╗ ██║       ██████╔╝    ██║    ████████╗ ██║  ╚██╗
   ╚═════╝  ╚═╝   ╚═╝ ╚═══════╝ ╚═╝       ╚═════╝     ╚═╝    ╚═══════╝
╚═╝   ╚═════════╝ ╚═══╝ ╚═══╝
                            https://www.jhipster.tech
Welcome to JHipster v7.9.3

□ Welcome to the JHipster Project Name □
? What is the base name of your application? multiple-human-readable-foreign-key-
fields
? What is the project name of your application? Multiple Human Readable Foreign Key
Fields Application
? Do you want to generate a local blueprint inside your application? No
? Which sub-generators do you want to override? cypress, entity-client
? Comma separated additional sub-generators.
? Add a cli? Yes
? Is cypress generator a side-by-side blueprint? Yes
? Is cypress generator a cli command? No
? What task do you want do implement at cypress generator? initializing
? Is entity-client generator a side-by-side blueprint? Yes
? Is entity-client generator a cli command? No
? What task do you want do implement at entity-client generator? initializing
? What is the default indentation? 2
   create .prettierrc.yml
   create package.json
   create .eslintrc.json
```

```
    force .yo-rc.json
   create .mocharc.cjs
   create README.md
   create test/utils.mjs
   create cli/cli.mjs
   create .github/workflows/generator.yml
   create .prettierignore
   create .gitignore
   create .gitattributes
   create .editorconfig
   create generators/cypress/index.mjs
   create generators/cypress/generator.spec.mjs
   create generators/cypress/generator.mjs
   create generators/entity-client/generator.mjs
   create generators/entity-client/index.mjs
    force .yo-rc.json
```

2. If you see the following error, there is an easy fix below:

```
Error [ERR_PACKAGE_PATH_NOT_EXPORTED]: Package subpath './lib/util/namespace' is
not defined by "exports" in ~/workspace/generator-jhipster-multiple-human-readable-
foreign-key-fields/node_modules/generator-jhipster/node_modules/yeoman-
environment/package.json
    at new NodeError (node:internal/errors:399:5)
    at exportsNotFound (node:internal/modules/esm/resolve:266:10)
    at packageExportsResolve (node:internal/modules/esm/resolve:602:9)
    at resolveExports (node:internal/modules/cjs/loader:550:36)
    at Module._findPath (node:internal/modules/cjs/loader:619:31)
    at Module._resolveFilename (node:internal/modules/cjs/loader:1046:27)
    at Module._load (node:internal/modules/cjs/loader:905:27)
    at Module.require (node:internal/modules/cjs/loader:1127:19)
    at require (node:internal/modules/helpers:112:18)
    at Object.<anonymous> (/Users/amarppatel/workspace/generator-jhipster-multiple-
human-readable-foreign-key-fields/node_modules/generator-
jhipster/utils/blueprint.js:19:25)
    at Module._compile (node:internal/modules/cjs/loader:1246:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1300:10)
    at Module.load (node:internal/modules/cjs/loader:1103:32)
    at Module._load (node:internal/modules/cjs/loader:942:12)
    at Module.require (node:internal/modules/cjs/loader:1127:19)
    at require (node:internal/modules/helpers:112:18)
    at Object.<anonymous> (/Users/amarppatel/workspace/generator-jhipster-multiple-
human-readable-foreign-key-fields/node_modules/generator-jhipster/cli/environment-
builder.js:27:82)
    at Module._compile (node:internal/modules/cjs/loader:1246:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1300:10)
    at Module.load (node:internal/modules/cjs/loader:1103:32)
    at Module._load (node:internal/modules/cjs/loader:942:12)
    at Module.require (node:internal/modules/cjs/loader:1127:19)
    at require (node:internal/modules/helpers:112:18)
    at Object.<anonymous> (/Users/amarppatel/workspace/generator-jhipster-multiple-
human-readable-foreign-key-fields/node_modules/generator-
jhipster/cli/program.js:26:28)
    at Module._compile (node:internal/modules/cjs/loader:1246:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1300:10)
    at Module.load (node:internal/modules/cjs/loader:1103:32)
    at Module._load (node:internal/modules/cjs/loader:942:12)
    at ModuleWrap.<anonymous> (node:internal/modules/esm/translators:168:29)
    at ModuleJob.run (node:internal/modules/esm/module_job:193:25)
```

3. Add the line "./lib/util/namespace": "./lib/util/namespace.js" to the local project yoeman-environment package.json at the end of the "exports" section as follows. You will need to add a "," after "./package.json".

```
vi ~/workspace/generator-jhipster-multiple-human-readable-foreign-key-
fields/node_modules/generator-jhipster/node_modules/yeoman-environment/package.json
```

4. This is what the package.json should look like:

```
  "exports": {
    ".": "./lib/environment.js",
    "./cli/": "./cli/",
    "./lib/": "./lib/",
    "./lib/util/": "./lib/util/",
    "./adapter": "./lib/adapter.js",
    "./conflicter": "./lib/util/conflicter.js",
    "./log": "./lib/util/log.js",
    "./transform": "./lib/util/transform.js",
    "./package.json": "./package.json",
    "./lib/util/namespace": "./lib/util/namespace.js"
  },
```

5. Then run the blueprint command again, to finish generating the blueprint

```
cd ~/workspace/generator-jhipster-multiple-human-readable-foreign-key-fields
jhipster generate-blueprint
```

6. You should see success message now:

```
Application successfully committed to Git from ~/workspace/generator-jhipster-
multiple-human-readable-foreign-key-fields.
Congratulations, JHipster execution is complete!
Sponsored with ❤️  by @oktadev.
```

Use the Blueprint Locally to Generate Code

1. Link the blueprint locally

```
cd ~/workspace/generator-jhipster-multiple-human-readable-foreign-key-fields
npm link
```

Modify the Blueprint to Enable Multiple Human-readable Foreign key Fields.

1. Added the following files:

```
generators/constants-saathratri.js
generator/utils-saathratri.js
generators/entity-client/templates/angular/src/main/webapp/app/entities/entity.model.ts.ejs
generators/entity-client/templates/angular/src/main/webapp/app/entities/detail/entity-management-detail.component.html.ejs
generators/entity-client/templates/angular/src/main/webapp/app/entities/list/entity-management.component.html.ejs
generators/entity-client/templates/angular/src/main/webapp/app/entities/update/entity-management-update.component.html.ejs
```

Have Fun with Creating JHipster Blueprints!

I hope you enjoyed this demo, and it helped you understand how to build blueprints with JHipster.

□□ Find the code on GitHub: https://github.com/amarpatel-xx/generator-jhipster-multiple-human-readable-foreign-key-fields

□ Read the README.md post at https://github.com/amarpatel-xx/jhipster-multiple-human-readable-foreign-key-fields-example to see a working example of a microservices architecture using an Angular client that was generated using this blueprint. You will see how multiple foreign key fields are displayed when a relationship's fields are shown.