Development Guide

Table of contents

1. Technologies and Versions

Project is created with:

- Java (11.0.6)
- Node (v10.19.0) and npm (6.13.7)
- Gradle (6.1.1)
- Angular CLI (1.7.4)
- REST
- SpringBoot
- JsonDB (http://jsondb.io/)
- Styling TodoMVC CSS (https://github.com/tastejs/todomvc-app-css)

With the script `bin/check-version.sh` you can check your installations and versions.

2. Technical configuration

Formatter: google-java-format (installation via Plugin for IDE)
Local run, Swagger is running under: localhost:8080/swagger-ui.html
Use SonarLint (IDE Plugin) for code quality checks.

2.1. Technical configuration

2.1.1. Windows with Ubuntu Bash

- Java (11.0.6)

```
apt-get install default-jdk
```

- Node (v10.19.0) and npm (6.13.7):

```
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install nodejs
```

- Angular CLI (1.7.4)

```
 npm install -g @angular/cli@1.7.4
```

Gradle (6.1.1)

```
sudo apt install gradle
```

(or use *./gradlew* instead of *gradle*)

3. Version-Control-System

3.1. Commit Rule

The commit message must always adhere to the following structure

```
#111 commit-message
```

The number should be the ticket number.

Smart would be this rule:<br> Format: `#<ticket-number> - <type>: <subject>`

Example:

```
#111 - feat: add hat wobble
^--^   ^--^  ^------------^
|         |     |
|         |     +-> Summary in present tense.
|         |
|         +-------> Type: chore, docs, feat, fix, refactor, style, or test.
+-> ticket-number
```

- `feat`: (new feature for the user, not a new feature for build script)
- `fix`: (bug fix for the user, not a fix to a build script)
- `docs`: (changes to the documentation)
- `style`: (formatting, missing semi colons, etc; no production code change)
- `refactor`: (refactoring production code, eg. renaming a variable)
- `test`: (adding missing tests, refactoring tests; no production code change)
- `chore`: (updating grunt tasks etc; no production code change)

3.2. Branch Rule

```
typ/123_some_text
```

- `typ`: feature or fix
- `123_some_text`: summary in present tense

4. IDE

4.1. IntelliJ

The following plugins might be useful:

- Git Conflict
- GitToolBox
- google-java-format
- JRebel
- Maven Helper
- SonarLint
- Spotbugs
- Spring Tools
- Spring Boot
- Save Actions

4.1.1. Start in IntelliJ

You can choose "Spring Boot" and then configure:

- Main class: com.thieme.doctor.Application
- Active profiles: mock (or the profile you want)

4.1.2. Start in IntelliJ Community

Start the Application as normal Java Application

4.2. Eclipse

…

4.3. VS Code

Extensions:

- Angular Essentials
- Angular Snippets
- Auto Import
- Code Spell Checker
- CSS Intellisense

- CSS Peek
- Debugger for Firefox
- Debugger for Chrome
- ESLint
- Git Blame
- Git Diff and Merge Tool
- Git Extension Pack
- Git Graph
- Git History
- Git History Diff
- Git Merger
- Git Tree Compare
- gitignore
- GitLens
- HTML CSS Support
- IntelliJ IDEA Keybindings
- IntelliSense for CSS class names in HTML
- JavaScript (ES6) code snippets
- js-intellisense
- LintLens
- Live Preview
- Live Sass Compiler
- Local History
- Markdown All in One
- Markdown Editor
- Material Icon THeme
- Path Intellisense
- Peacock
- Prettier
- Preview
- Rainbow Brackets
- SonarLint
- StyleLint
- Todo Tree
- TypeScript Importer

- vsPrettier
- vscode-icons
- XML Tools
- YAML

5. Collaboration

- One-man project

6. Definition of Done

6.1. Feature/User Story

- Acceptance criteria are met
- Refactoring is complete
- Code builds with no error
- Unit tests are written and pass
- Existing Unit Tests pass
- Software is runnable correctly
- Sufficient diagnostics/telemetry are logged
- UX review is complete (if applicable)
- Documentation is updated
- Lint Tools analyzed the Code and new issues are fixed
- Code review is complete
- The feature is merged into the develop branch
- Ticket for the code is pushed in Jira to test row

6.2. Sprint Goal

- Definition of Done for all user stories included in the sprint are met
- Product backlog is updated
- Functional and Integration tests pass
- Performance tests pass
- End 2 End tests pass
- All bugs are fixed
- The sprint is signed off from developers, software architects, project manager, product owner etc.

6.3. Release/Milestone

- Code Complete (goals of sprints are met)
- Release is marked as ready for production deployment by product owner

7. FAQ

1. **Question**
    - *Answer*

8. References

- …

9. Glossary

```json
{
  "glossary": [
    {
      "domain-term": "Task",
      "technical-term": "task",
      "description": "TODO-element"
    }
  ]
}
```