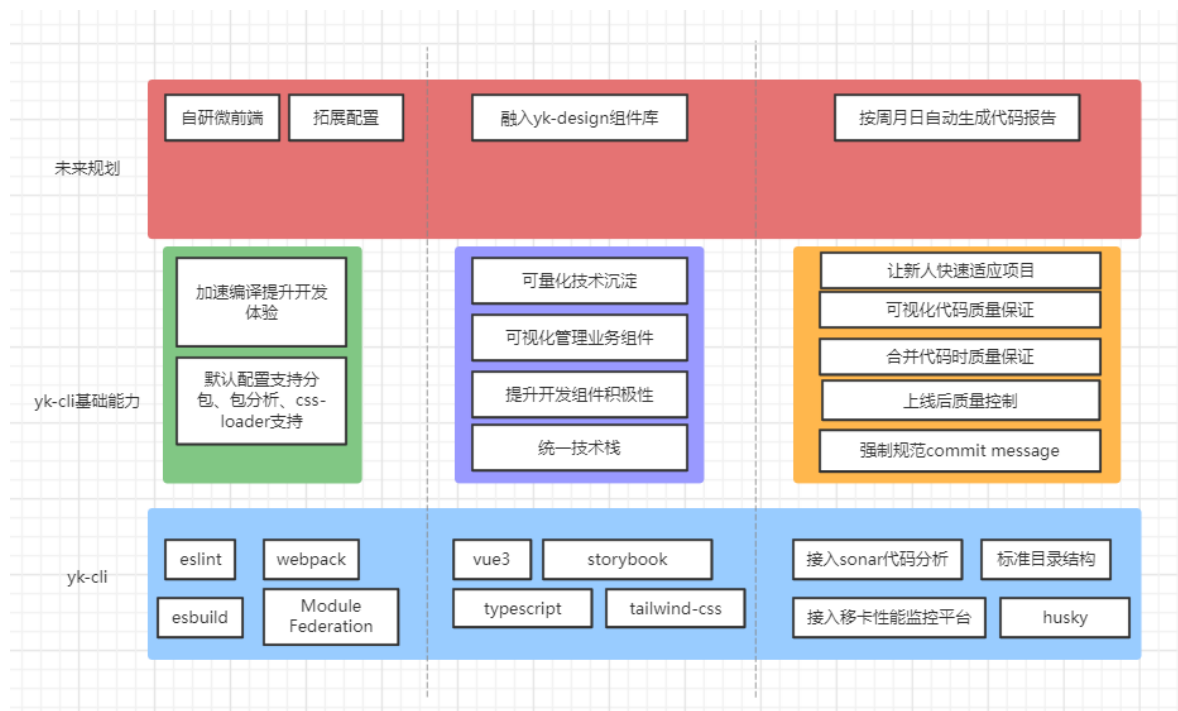


yeahka-cli

An easy way to start a Yeahka project



本地测试

环境准备

1. 登录vpn
2. 推荐使用nrm进行包管理

```
~ nrm add yeahka http://registry.npm.lepass.cn/

add registry company success
nrm ls
nrm use yeahka
```

3. 切换node版本至14以上，包括14，建议本地直接安装node14

本地测试

安装pnpm

```
npm install -g pnpm
```

依次在命令行执行一下代码：

```
git clone http://gitlab.yeahka.com/web/yeahka-cli.git
cd ./yeahka-cli
git fetch --all
git checkout develop
pnpm install
cd ./playground
node ../playground.js
```

线上版本验证

```
nrm use yeahka-cli
npx yeahka-cli
```

基础能力-创建模板项目

```
npx yeahka-cli
```

实现注入式js项目的基础

npx

允许你从本地或者远端的npm包中运行一段js代码

简而言之，只需要我们将一段脚本命令上传到移卡的私仓，再用npx运行这段代码，就可以生成预设好的模板项目了。

模板项目的组成

base

模板项目的基础

code

模板项目本身的代码，包括src中的示例vue code，如hello world页面等。配置router时需要添加router配置，配置测试文件需要测试代码等。

config

模板项目的配置文件，核心由两部分组成，package.json以及vite.config.js。修改这两个文件足以统一一个项目的配置

entry

入口文件，配置mainjs，编写入口函数，如router全局注册的方法，createApp的挂载等。

pre-commit校验

一般来说，commit message 应该清晰明了，说明本次提交的目的，如图：

Commits on Jan 5, 2016



docs(error/\$rootScope/inprog): add missing "\$timeout" ...

wytesk133 committed with petebacondarwin 10 days ago



docs(tutorial/2): add e2e test missing filename ...

monkpit committed with petebacondarwin 3 hours ago



revert: feat(\$compile): Allow ES6 classes as controllers with `bindTo`... ...

petebacondarwin committed 3 hours ago



fix(\$animateCss): only (de)register listeners when events have been a... ...

Narretz committed 25 days ago



fix(loader): use `false` as default value for `transclude` in compone... ...

sarod committed with petebacondarwin 18 days ago

Commits on Jan 3, 2016



feat(\$compile): Allow ES6 classes as controllers with `bindToControll... ...

lgalfaso committed 21 days ago



chore(*): Updated year in licence ...

leuanG committed with lgalfaso 3 days ago

Commits on Jan 1, 2016



docs: reorganize information about interpolation ...

Narretz committed on Dec 6, 2015



docs: add docs for ngPattern, ngMinlength, ngMaxlength ...

Narretz committed on Sep 24, 2015



docs(\$interpolateProvider): remove superfluous ng-app attribute ...

Narretz committed 4 days ago

所以在这里使用了husky对commit message进行强制的约束。

Header部分只有一行，包括三个字段：`type`（必需）、`scope`（可选）和`subject`（必需）。

(1) type

`type`用于说明 commit 的类别，只允许使用下面7个标识。

- feat: 新功能 (feature)
- fix: 修补bug
- docs: 文档 (documentation)
- style: 格式 (不影响代码运行的变动)
- refactor: 重构 (即不是新增功能，也不是修改bug的代码变动)
- test: 增加测试
- chore: 构建过程或辅助工具的变动

如果 `type` 为 `feat` 和 `fix`，则该 commit 将肯定出现在 Change log 之中。其他情况（`docs`、`chore`、`style`、`refactor`、`test`）由你决定，要不要放入 Change log，建议是不要。

(2) scope

`scope`用于说明 commit 影响的范围，比如数据层、控制层、视图层等等，视项目不同而不同。

(3) subject

`subject`是 commit 目的的简短描述，不超过50个字符。

- 以动词开头，使用第一人称现在时，比如 `change`，而不是 `changed` 或 `changes`
- 第一个字母小写
- 结尾不加句号 (.)

参考: https://www.ruanyifeng.com/blog/2016/01/commit_message_change_log.html

在这些文件夹中配置好预设的模板选项，最后就可以组合成一个完整的项目了，这样yeahka-cli就有了它的基础能力-配置预设项目的能力。

进阶能力-自动化项目信息

在设想的架构图中，我们的yeahka-cli，不单单只有模板项目的功能，还有一些进阶的功能，如自动同步公共组件，sonar代码分析，按时间维度生成项目报告等。

以下服务均建立在test分支上。

front-end-bi

前端测试服中会架设一台nestjs为基础的服务，以nginx做反向代理，访问到/front-end-bi时即为访问平台服务，该nestjs服务将接管/front-end-bi/*的访问，一站式的建立后端服务（对项目的增删改查）以及可视化平台页面/storybook平台的前端页面。

项目管理

进阶能力不属于项目范畴内，所以不需要在项目模板中添加或者修改内容，我们需要一张表，帮我们记录每个项目这些内容。

1. 项目名称 (pid)
2. 项目产出的公共组件
3. 项目gitlab链接
4. 项目代码质量

项目名称对应了性能监控平台中的pid，在新建项目的同时会对前端项目bi平台发送一条请求，建立一条数据。（已有项目目前考虑手动添加，或者考虑加一种注入式更新npxyeahka-cli --> 生成法）

代码质量由sonar进行分析管理。

组件管理

通过jenkins的更新钩子，在前端项目的test分支上更新时自动触发，同时更新项目的storybook，将该项目的storybook产出至/front-end-bi/yeahka-storybook目录下，即可实现同时更新。

发布的时候storybook build并将生成文件scp至yeahka-storybook中项目目录名下

/wap/yeahka-storybook目录结构为：

```
├── okr
|   ├── index.html
├── linktime
|   ├── index.html
└── index.html
```

整体设计

1. storybook汇总各个项目中组件
2. sonar扫描

3. 自动接入性能监控平台 以新建project sample为例

