

1. 安装

<http://mpvue.com/mpvue/>

常见问题 <http://mpvue.com/qa/>

```
# 全局安装 vue-cli
$ rpm install --global vue-cli

# 创建一个基于 mpvue-quickstart 模板的新项目
$ vue init mpvue/mpvue-quickstart my-project

# 安装依赖
$ cd my-project
$ npm install
# 启动构建
$ npm run dev
```

注意：我们安装的vue cli 3 是不支持vue init 功能，需要以下操作

Vue CLI >= 3 和旧版使用了相同的 `vue` 命令，所以 Vue CLI 2 (`vue-cli`) 被覆盖了。如果你仍然需要使用旧版本的 `vue init` 功能，你可以全局安装一个桥接工具：

```
npm install -g @vue/cli-init
# `vue init` 的运行效果将会跟 `vue-cli@2.x` 相同
vue init webpack my-project
```

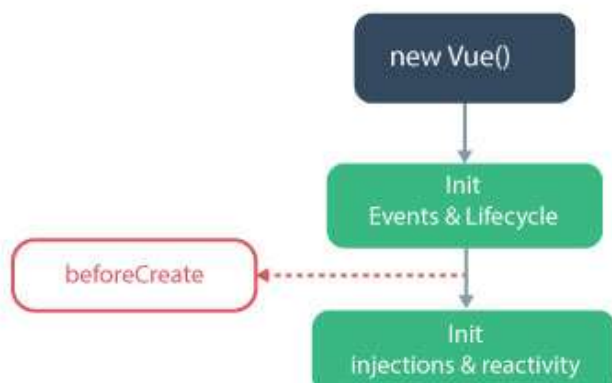
sh

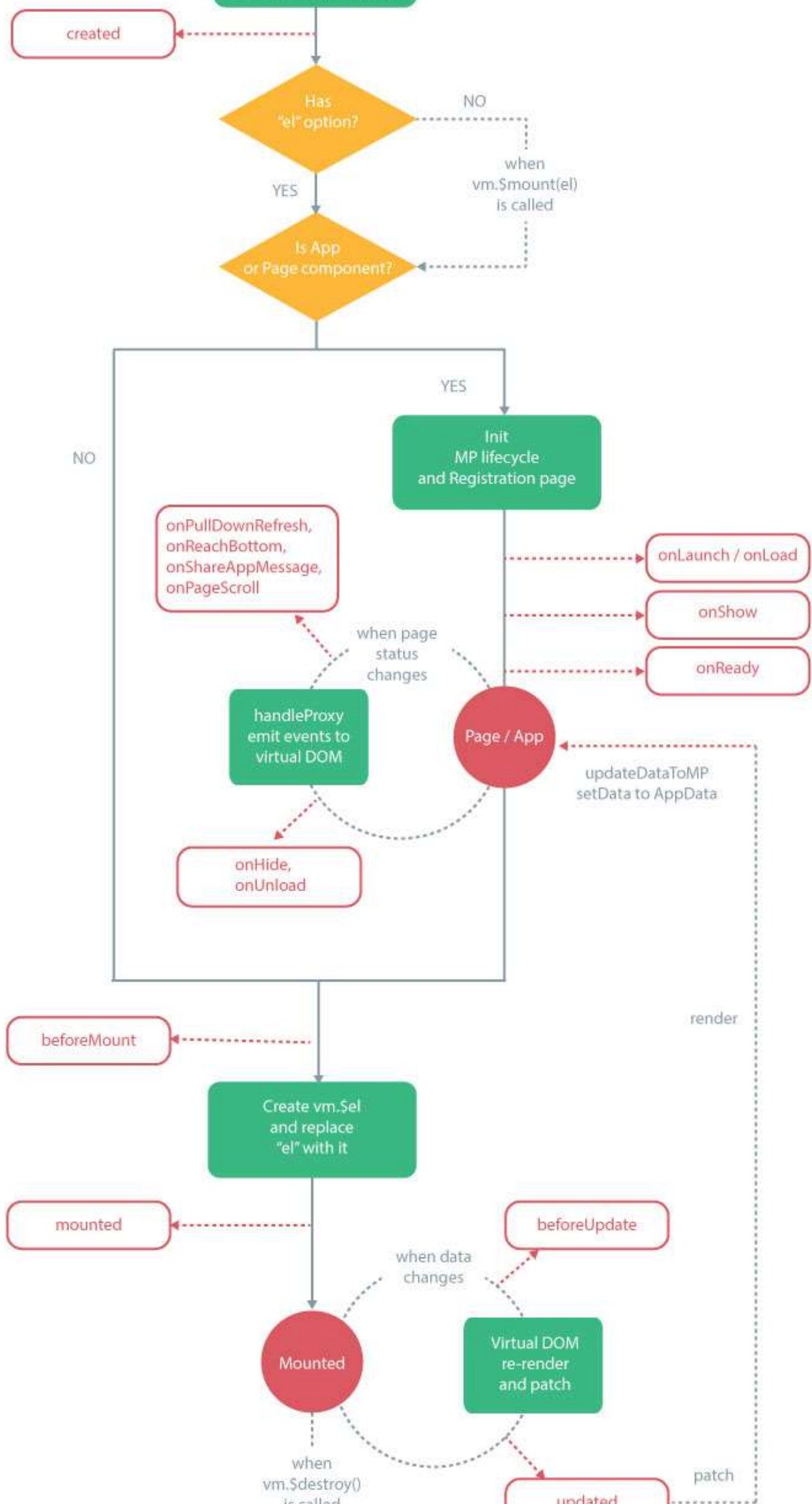
2. mpvue框架原理

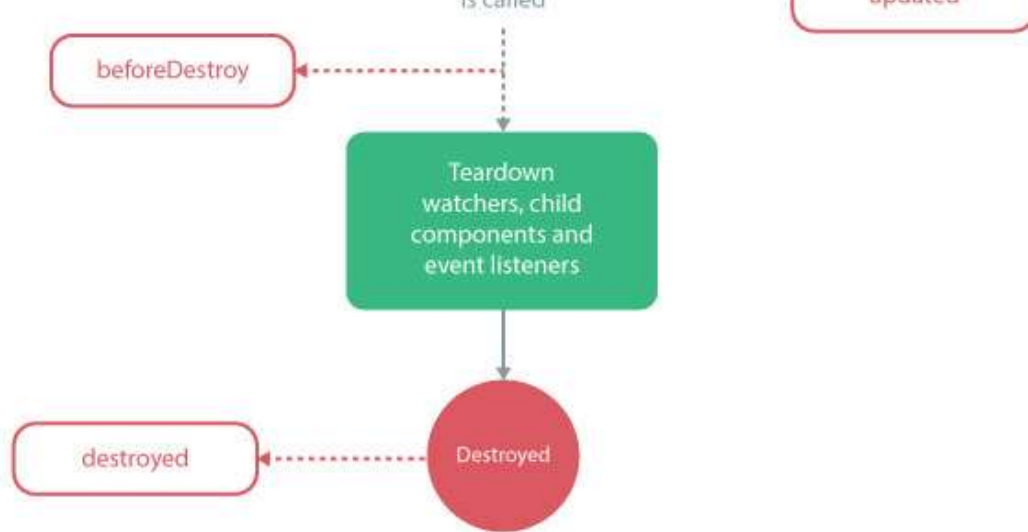
- mpvue 保留了 vue.runtime 核心方法，无缝继承了 Vue.js 的基础能力
- mpvue-template-compiler 提供了将 vue 的模板语法转换到小程序的 wxml 语法的能力
- 修改了 vue 的建构配置，构建出符合小程序项目结构的代码格式： json/wxml/wxss/js 文件

3. 生命周期

<http://mpvue.com/mpvue/#vue>







* Runtime-only, no template compilation

4. vue在小程序开发中的应用与区别

- 模板语法
- 计算属性
- 不支持函数
- Class 与 Style 绑定
- 条件渲染
- 列表渲染
- 事件处理器
- 表单控件绑定
- 组件

5. 常见问题

(1) 如何获取小程序在 page onLoad 时候传递的 options

在所有 页面 的组件内可以通过 `this.$root.$mp.query` 进行获取。

(2) 如何获取小程序在 app onLaunch/onShow 时候传递的 options

在所有的组件内可以通过 `this.$root.$mp.appOptions` 进行获取。

(3) 如何捕获 app 的 onError

由于 onError 并不是完整意义的生命周期，所以只提供一个捕获错误的方法，在 app 的根组件上添加名为 onError 的回调函数即可。如下：

```
export default {
  // 只有 app 才会有 onLaunch 的生命周期
  onLaunch () {
    // ...
  },

  // 捕获 app error
  onError (err) {
    console.log(err)
  }
}
```

(4) vue-router 支持吗？

路由是不能支持的，因为小程序无法动态的插入和控制节点，几乎无法实现。而且小程序对页面的切换性能做了很多优化，页面切换体验也好了很多，所以使用 vue-router 也不那么必要

(5) 为什么我新增了页面，没有反应？

因为 webpack 编译的文件是由配置的 entry 决定的，新增的页面并没有添加进 entry，所以需要手动 `npm run dev` 一下，考虑不是高频操作，所以还可以忍受

(6) 能不能引用第三方的 UI 库？

原理上是可以的，但是要去掉 Dom 和 Bom 相关的 API 操作，还要去掉 mpvue 不支持的一些 vue 特性，小程序不支持的 css 选择器等，去掉这些就可以使用了，欢迎大家贡献 mpvue 的 UI 组件库。

(7) 单文件组件中 style 支持 scss 写法吗？

`cnpm install --save sass-loader@7.1.0` (sass-loader 8.* 不兼容)