Table of Contents

[Goal 1](#_Toc453084585)

[Design 1](#_Toc453084586)

[Thought 1](#_Toc453084587)

[Plan 1](#_Toc453084588)

[Tasks 1](#_Toc453084589)

[Project structure 2](#_Toc453084590)

[Vision 2](#_Toc453084591)

[Environment 2](#_Toc453084592)

[Tool stack 2](#_Toc453084593)

[Dependencies 3](#_Toc453084594)

[Modules 3](#_Toc453084595)

[Project Interface 3](#_Toc453084596)

[Program procedure 3](#_Toc453084597)

[Core data structure 3](#_Toc453084598)

[core algorithm 3](#_Toc453084599)

[Project details 3](#_Toc453084600)

[Knowledge 3](#_Toc453084601)

[Style 3](#_Toc453084602)

[Tricks 3](#_Toc453084603)

[evaluation 3](#_Toc453084604)

[advantage 3](#_Toc453084605)

[disadvantage or risk 3](#_Toc453084606)

[scenes 3](#_Toc453084607)

## Goal

Understand the project node, from whole to part.

## Design

### Thought

## Plan

### Tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **module** | **task** | | **description** | **time** | **status** |
| **Project structure** | Vision | | Project vision |  |  |
| Environment | | Running environment |  |
| Tool stack | | Dev tool, test tool, deploy tool.  (a tool stack flow and a function table) |  |
| Dependencies | | Find all the dependents of the project, and know those dependencies’ functions.  (dependency chart and a function table) |  |  |
| Modules | | Distinct modules and find the relationship of all modules.  (dependency chart and a function table) |  |  |
| **Program procedure** | ${module} | Module interface | Interfaces of module, like API, UI.  ( an interface table) |  |  |
| Core data structure | The core data structures used in the project. |  |  |
| Core algorithm | The core data algorithms used in the project. |  |  |
| **Project details** | ${module} | Knowledge | The knowledge you do not know |  |  |
| Style | Style of project: comment, naming, lint, closure. Slips, law. |  |  |
| Tricks | Programing tricks. |  |  |
| Other |  |  |  |
| **evaluation** | Advantage | | The advantages of this project |  |  |
| Disadvantage | | The disadvantage of this project |
| Scenes | | The scenes it suits and not suits |

## Project structure

### Vision

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

### Environment

Unix

 gcc and g++ 4.8 or newer, or

 clang and clang++ 3.4 or newer

 Python 2.6 or 2.7

 GNU Make 3.81 or newer

<https://github.com/nodejs/node/blob/master/BUILDING.md>

### Tool stack

### Dependencies

|  |  |
| --- | --- |
| **Module name** | **Function description** |
|  |  |

### Modules

|  |  |
| --- | --- |
| **Module name** | **Function description** |
|  |  |

### Project Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **interface** | **description** | **invoke** | **parameters** | | **result** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Program procedure

### Core data structure

### core algorithm

## Project details

### Knowledge

|  |  |
| --- | --- |
| **target** | **method** |
|  |  |

### Style

### Tricks

## evaluation

### advantage

### disadvantage or risk

### scenes