

# 项目申请书

## 一、项目信息

- 项目ID: 2005027
- 项目名称: 命令行 Dashboard
- 所属社区: Apache SkyWalking
- 项目描述: 为 SkyWalking CLI 提供一个统一的 dashboard 视图, 可以查看全局数据
- 产出要求:
  - swctl dashboard 命令
  - dashboard 包含 global 相关的指标
  - dashboard 自动刷新
- 仓库地址: <https://github.com/apache/skywalking-cli>

## 二、技术方案

### 2.1 设计

首先, 分析参考web ui的[dashboard](#), 确认所需要的具体数据, 参考项目 `assets/graphqls` 目录下其他命令的 `graphql` 文件格式, 编写 `dashboard` 命令要请求数据的 `graphql` 文件。

然后将整个 `dashboard` 要展示全局信息划分为三个部分:

- 第一部分为展示服务和端点相关信息, 可使用 `termui` 库的 `Gauge` 组件实现。
- 第二部分为展示全局响应延迟信息, 可以复用项目 `display/graph/linear` 目录下的代码, 使用多条折线图来显示。
- 第三部分为展示全局信息, 可以复用项目 `display/graph/heatmap` 目录下的代码, 使用热力图来显示。

为了之后便于扩展命令, 可将命令设计为:

```
swctl dashboard global [--refresh-time=time]
```

参数含义如下所示:

argument	description	default
<code>--refresh-time</code>	the refresh interval (s)	6

## 2.2 测试

根据官方文档，使用 `docker` 搭建 `SkyWalking` 虚拟环境，用于调试和测试代码。

## 2.3 文档

仿照README文件内容，添加 `dashboard` 命令的说明以及示例，并在PR中附上演示视频。

# 三、时间规划

将时间划分为三个大阶段，每个大阶段含有若干个以周为单位的小时间段：

## 3.1 熟悉阶段

本阶段的主要任务是熟悉项目代码以及涉及的相关知识。

时间段	事项
7 月 1 日 -- 7 月 7 日	通读项目源代码以及文档；复习 <code>Golang</code> 和 <code>GraphQL</code> 相关知识
7 月 8 日 -- 7 月 14 日	搭建 <code>SkyWalking</code> 环境，用于之后调试与测试

## 3.2 实施阶段

本阶段的主要任务是编写代码，实现产出要求。

时间段	事项
7 月 15 日 -- 7 月 21 日	实现第一部分全局信息的展示
7 月 22 日 -- 7 月 28 日	实现第二部分全局信息的展示
7 月 29 日 -- 8 月 4 日	实现第三部分全局信息的展示
8 月 5 日 -- 8 月 11 日	完成中期报告并提交
8 月 12 日 -- 8 月 18 日	实现按照给定参数自动刷新

## 3.3 收尾阶段

本阶段的主要任务是测试、调试代码，完善文档。

时间段	事项
8 月 19 日 -- 8 月 25 日	测试、调试代码
8 月 26 日 -- 9 月 1 日	完善文档，提交PR
9 月 2 日 -- 9 月 8 日	完成结项报告并提交