

# 项目健康监控实施规范文档

## 健康度探(Functional Health Monitor)

监控目的：深度探测后端依赖（数据库、Redis、磁盘空间、关键业务逻辑）。

### 1.1 后端接口逻辑 (FastAPI 示例)

这个接口会检查所有核心组件。只要有一个组件失败，根节点的 `result` 就会变为 `fail`。

Python

代码块

```
1  from fastapi import FastAPI, Response, status
2  import shutil # 用于磁盘检查
3
4  app = FastAPI()
5
6  @app.get("/api/health")async def health_check(response: Response):
7      # --- 1. 检查数据库 ---
8      db_ok = await check_db_connection() # 假设返回 True/False
9      # --- 2. 检查 Redis ---
10     redis_ok = await check_redis_status() # 假设返回 True/False
11     # --- 3. 检查磁盘空间 ---
12     total, used, free = shutil.disk_usage("/")
13     disk_ok = (free / total) > 0.1 # 剩余空间需大于 10%
14     # --- 4. 汇总总开关 ---
15     all_systems_go = db_ok and redis_ok and disk_ok
16
17     # 构建详细数据
18     health_data = {
19         "result": "succeed" if all_systems_go else "fail",
20         "timestamp": "2026-02-10T16:05:00Z",
21         "details": {
22             "database": {"result": "succeed" if db_ok else "fail"},
23             "redis": {"result": "succeed" if redis_ok else "fail"},
24             "disk": {"result": "succeed" if disk_ok else "fail", "free_gb": f"
25                 {free // (2**30)}GB"}
26         }
27     }
28
29     # 如果有任何失败，返回 503 状态码
30     if not all_systems_go:
31         response.status_code = status.HTTP_503_SERVICE_UNAVAILABLE
```

```
31  
32     return health_data
```

## 2. 接口概览

- **接口路径:** /api/health
- **请求方法:** GET
- **核心逻辑:** 汇总数据库、Redis 及磁盘空间的状态。只有全部组件正常，接口才认为服务整体可用。

这是接口逻辑的核心。下表说明了不同组件状态如何影响最终的响应。

组件名称	正常标准 (succeed)	异常标准 (fail)	故障后果
Database	数据库连接池可正常获取连接	无法连接、超时、或鉴权失败	关键故障：导致 result 变为 fail
Redis	Ping 命令返回 PONG	连接超时或连接数满	关键故障：导致 result 变为 fail
Disk Space	根目录剩余空间 > 10%	剩余空间 ≤ 10%	关键故障：导致 result 变为 fail

## 2. 响应状态码与内容

接口根据内部组件的汇总结果，返回不同的 HTTP 状态码，以便监控软件（如 Uptime Kuma）快速识别。

业务状态	HTTP 状态码	JSON result 字段	监控系统表现
系统全线正常	200 OK	"succeed"	<span style="color: green;">● 正常 (UP)</span>
任一组件故障	503 Service Unavailable	"fail"	<span style="color: red;">● 报警 (DOWN)</span>

## 4. JSON 返回示例

### A. 正常场景 (HTTP 200)

JSON

```
1  {
2    "result": "succeed",
3    "timestamp": "2026-02-10T16:05:00Z",
4    "details": {
5      "database": { "result": "succeed" },
6      "redis": { "result": "succeed" },
7      "disk": { "result": "succeed", "free_gb": "45GB" }
8    }
9 }
```

## B. 异常场景 (HTTP 503)

例如 Redis 挂掉时的返回：

JSON

代码块

```
1  {
2    "result": "fail",
3    "timestamp": "2026-02-10T16:10:00Z",
4    "details": {
5      "database": { "result": "succeed" },
6      "redis": { "result": "fail" },
7      "disk": { "result": "succeed", "free_gb": "45GB" }
8    }
9 }
```