

# hydra

hydra 是基于 go 语言和众多开源项目实现的分布式微服务框架

hydra[ˈhaɪdrə]致力于提供统一的，丰富功能的后端开发框架，降低后端开发的复杂性，提高开发效率。目前已支持的服务类型有：http api 服务，rpc 服务，websocket 服务, mqc 消息消费服务，cron 定时任务服务, web 服务，静态文件服务。

## 特性

- 后端一体化框架, 支持 6+服务器类型
- 微服务的基础设施, 服务注册发现，熔断降级，集成监控与统一配置管理
- 多集群模式支持，对等，主备，分片等
- 丰富的后端库支持 redis,memcache,activieMQ,mqtt,influxdb,mysql,oracle,elasticsearch,jwt 等等
- 跨平台支持(linux,macOS 10.9+,windows 7+)
- 20+线上项目实践经验
- 全 golang 原生实现

## 示例

### 1. 编写代码

新建文件夹 hello ,并添加 main.go 文件,输入以下代码:

```

package main

import (
    "github.com/micro-plat/hydra/context"
    "github.com/micro-plat/hydra/component"
    "github.com/micro-plat/hydra/hydra"
)

func main() {
    app := hydra.NewApp(
        hydra.WithPlatName("myplat"), //平台名称
        hydra.WithSystemName("demo"), //系统名称
        hydra.WithClusterName("test"), //集群名称
        hydra.WithServerTypes("api"), //服务器类型为http api
        hydra.WithRegistry("fs://../"), //使用本地文件系统作为注册中心
        hydra.WithDebug())

    app.API("/hello", hello)
    app.Start()
}

func hello(ctx *context.Context) (r interface{}) {
    return "hello world"
}

```

## 2. 编译生成服务

```
go install hello
```

## 3. 安装本地服务和生成注册中心配置

```
hello install
```

## 4. 运行服务

```
./hello run
```

## 5. 测试服务

```
curl http://localhost:8090/hello
```

```
{"data":"hello world"}
```

以上代码可理解为:

1. 使用 文件系统 ( `fs://` )作为服务的注册中心和配置中心, `../` 作为服务的跟路径
2. 在注册中心创建 `/myplat/demo/api/test/` 节点作为服务的根路径
3. 将传入的 `hello` 函数作为 `api` 服务注册到服务器
4. 执行服务 `http://host:port/hello` 时执行服务 `func hello(ctx *context.Context) (r interface{})`
5. 可从 `*context.Context` 获取请求相关参数
6. `func hello` 的返回值作为当前接口的输出内容

执行 `hello install` 可理解为:

1. 安装配置数据,在注册中心创建节点 `/myplat/demo/api/test/` , 数据库配置 `/myplat/var/db/...` (当前未指定), 服务启动端口 `/myplat/demo/api/test/conf` (当前未指定启动端口,默认启动 `9090` ),当前示例采用了默认配置,未指定额外参数
2. 安装本地服务(后台运行服务,开机自动启动如: `systemd` 等等)
3. 安装后的服务配置可通过 `hello conf` 查看

执行 `hello run` 可理解为:

1. 连接注册中心( `fs://../` ),拉取服务配置,如: `/myplat/demo/api/test/conf/...` , `/myplat/var/...` 并监控 `/myplat/demo/api/test` 下所有配置的变化,变动后进行热更新
2. 启动服务器 `api` ,挂载注册的服务 `hello`
3. 将 `hello` 发布到注册中心 `/myplat/services/api/hello/providers`
4. 将当前服务器到监控目录 `/mysql/demo/api/test/servers/[ip:port]...`

## hydra微服务开发入门

### 服务配置与安装

### 服务注册

### 服务启动

### API 示例