# 目录四：controllers

```
export default {
```

```
urls: ['/schema', '/foo'],      // 指定url，未指定则以该文件路径进行推断
```

```
middlewares: ['schema-logger'], // 指定该次请求需执行的中间件，默认为`[]`
```

```
template: 'schema',                    // 指定HTML渲染的模板
```

```
return {foo: 'hello', bar: 'world'}
```

```
methods: ['GET'],                    // 指定method，默认为`["GET"]`
```

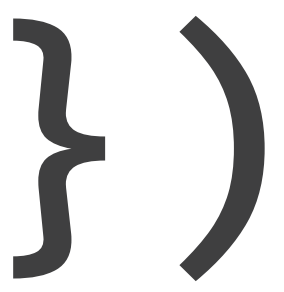
```
controller: async (ctx) ⇒ {    // Controller 函数 → 最后一个中间件
```

# Controller Schema 解析过程

✓ routes register

✓ init ctx.render() & run

```
app.use( async ctx ⇒ {
```



```
ctx.body = await template.toHtml()
```

```
const Template = ctx.$tpls[tpl]
```

```
const serveData = await controller.call(this, ctx, next)
```

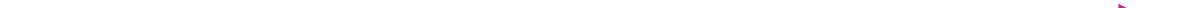
```
const template = new Template({ ctx, page, serveData, serveBundle: ctx.$bundles[page] })
```

```
middlewares: ['schema-logger'],
```

```
urls: ['/schema', '/foo'],
```

```
controller: async (ctx) ⇒ {
```

```
await ctx.render('schema', {foo: 'hello', bar: 'world'}, 'default')
```



```
controller: async (ctx) ⇒ {
```

```
await ctx.render('schema', {foo: 'hello', bar: 'world'}, 'default')
```