

# 基于 Serverless 的 Eggjs 后台管理系统配置 Session、创建 tools.js 服务、通过 svg-captcha 生成验证码

主讲教师：（大地）

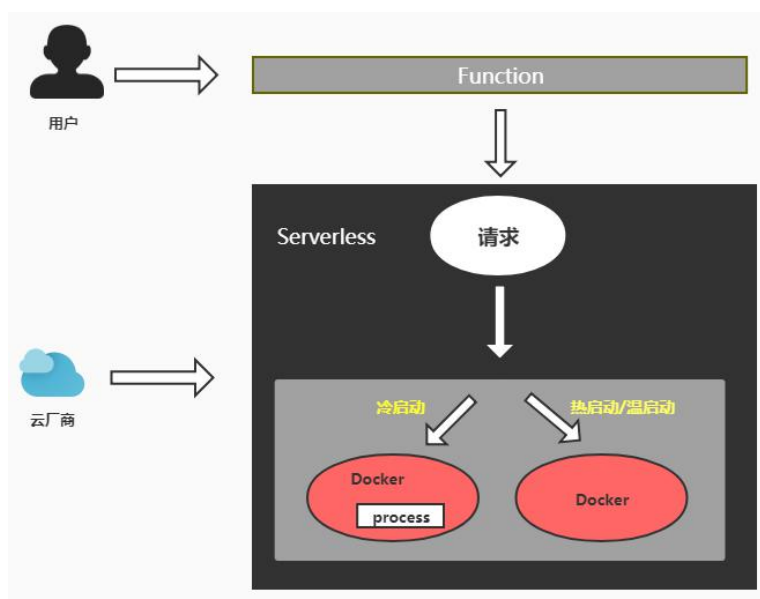
合作网站：[www.itying.com](http://www.itying.com) （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、 基于 Serverless 的 Egg.js 后台管理系统配置 Session.....	1
二、 配置 session.....	2
三、 svg-captcha 生成验证码.....	2

## 一、基于 Serverless 的 Egg.js 后台管理系统配置 Session

注意：Serverless 中使用 session 需要注意 Serverless 容器的冷启动和热启动。冷启动相比热启动稍微要慢一些，因为冷启动的时候需要创建容器。当我们的程序长时间没有人访问的时候运行我们程序的 Docker 容器会被销毁。Docker 容器销毁后重新访问我们的程序的时候又会执行冷启动。Docker 容器的重启会导致 session 丢失。所以我们在 Serverless 中使用 session 的时候要注意云厂商 Docker 容器默认销毁的时间。腾讯云的 Serverless 云函数如果 30 分钟内没有访问的话就会销毁容器。所以如果你的 session 要保存 30 分钟以上的话就需要把 session 存储到 redis 或者数据库中。





## 二、配置 session

<https://eggjs.org/zh-cn/core/cookie-and-session.html>

```
config.session={
  key:'SESSION_ID',
  maxAge:864000,
  httpOnly: true,
  encrypt: true,
  renew: true //延长会话有效期
}
```

设置:

```
this.ctx.session.userinfo={
  name:'张三',
  age:'20'
}
```

获取:

```
var userinfo=this.ctx.session
```

## 三、svg-captcha 生成验证码

生成验证码用到的模块: <https://www.npmjs.com/package/svg-captcha>

Koa 中使用:

```
var svgCaptcha = require('svg-captcha');
```

```
const captcha = svgCaptcha.create({
  size: 6,
  fontSize: 50,
  width: 100,
  height: 40,
  background: "#cc9966"
});
ctx.session.code = captcha.text;
ctx.response.type = 'image/svg+xml';
ctx.body = captcha.data;
```

**Egg.js** 中使用:

```
async captcha() {

  const captcha = svgCaptcha.create(
    {
      size: 4,
      fontSize: 50,
      width: 100,
      height: 40,
      background: "#cc9966"
    }
  );
  this.ctx.session.code = captcha.text;
  this.ctx.response.type = 'image/svg+xml';  /*指定返回的类型*/
  this.ctx.body = captcha.data;              /*给页面返回一张图片*/
}
```