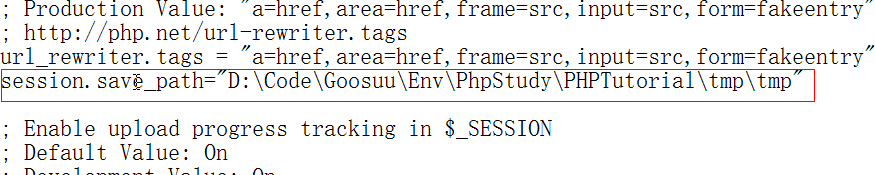
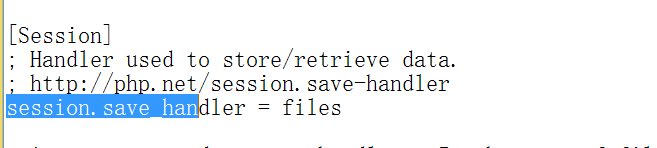
# Session会话技术

## PHP配置文件中session设置

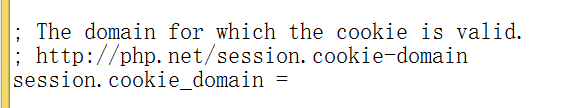
session文件存储位置



session数据的存储方式



Session cookie域

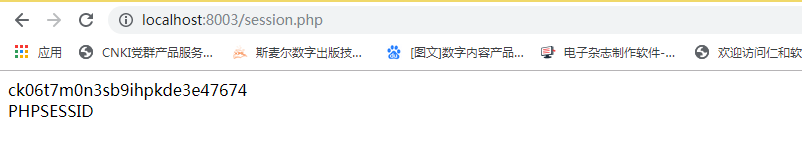


## 获取session\_id,session\_name示例

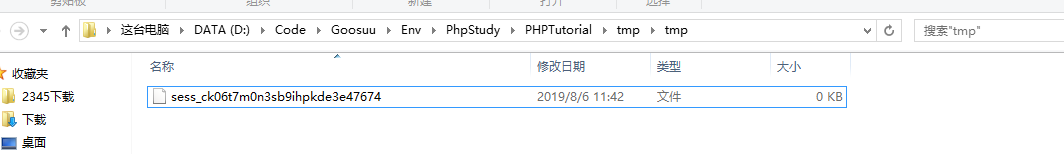
脚本

**function** show($msg){  
 **echo '<pre>'**;  
 *print\_r*($msg);  
 **echo '</pre>'**;  
 **echo '<br/>'**;  
}  
  
*# 开启session  
session\_start*();  
  
*# 获取session\_id()*show(*session\_id*());  
*# 设置session\_id  
session\_id*(**"lieyan123456"**);  
show(*session\_id*());  
  
*# 打印session\_name*show(*session\_name*());

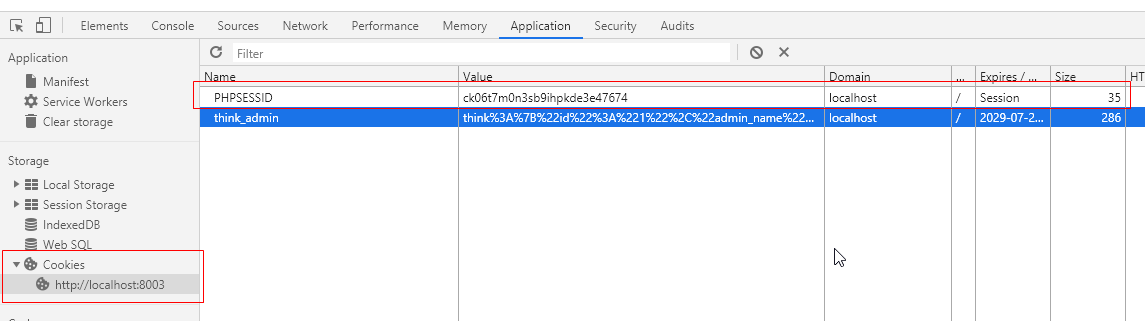
开启浏览器，访问session.php



查看服务器session存储位置，生成文件



浏览器cookie存储SESSION标识

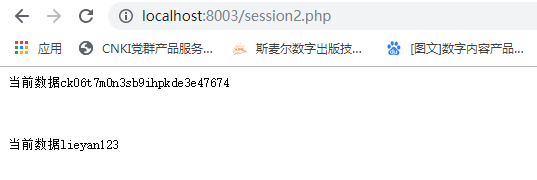


## 设置session数据，获取session数据示例

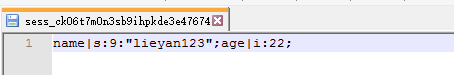
脚本：

*header*(**"Content-Type: text/html;charset=utf-8"**);  
**function** show($msg){  
 **echo '<pre>'**;  
 *print\_r*(**"当前数据"** . $msg);  
 **echo '</pre>'**;  
 **echo '<br/>'**;  
}  
  
*# 开启session  
session\_start*();  
  
*# 打印当前session*show(*session\_id*());  
  
*# 设置session数据*$\_SESSION[**'name'**] = **"lieyan123"**;  
$\_SESSION[**'age'**] = 22;  
  
*# 获取session数据*show($\_SESSION[**'name'**]);

开启浏览器，访问session2.php



服务器session文件保存session数据

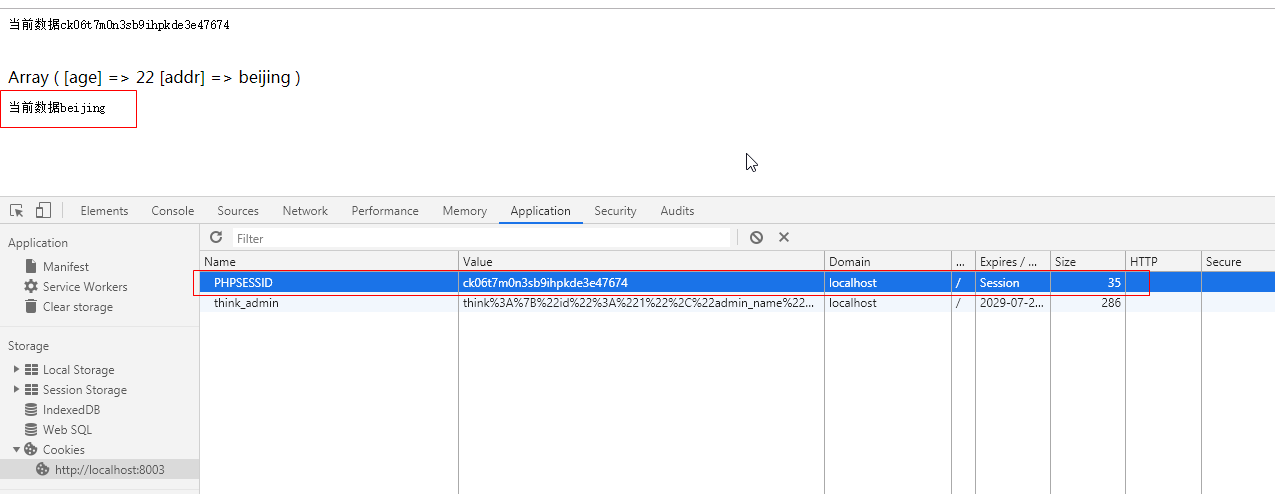


## 删除session数据、销毁session会话示例

脚本

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: goosuu  
 \* Date: 2019/8/6  
 \* Time: 13:43  
 \*/  
header*(**"Content-Type: text/html;charset=utf-8"**);  
**function** show($msg){  
 **echo '<pre>'**;  
 *print\_r*(**"当前数据"** . $msg);  
 **echo '</pre>'**;  
 **echo '<br/>'**;  
}  
  
*# 开启session  
session\_start*();  
  
*# 打印当前session*show(*session\_id*());  
  
*# 设置session数据*$\_SESSION[**'name'**] = **"lieyan123"**;  
$\_SESSION[**'age'**] = 22;  
$\_SESSION[**'addr'**] = **'beijing'**;  
  
  
*# 删除name***unset**($\_SESSION[**'name'**]);  
*print\_r*($\_SESSION);  
**echo '<br/>'**;  
  
*# 清空session数据  
# 方式1  
# session\_unset();  
# 方式2  
/\*$\_SESSION = array();  
print\_r($\_SESSION);  
echo '<br/>';\*/  
  
# 销毁session  
session\_destroy*();  
*# 销毁session，但是不会马上销毁内存中的session变量，当下次访问时，session才清空*show($\_SESSION[**'addr'**]);

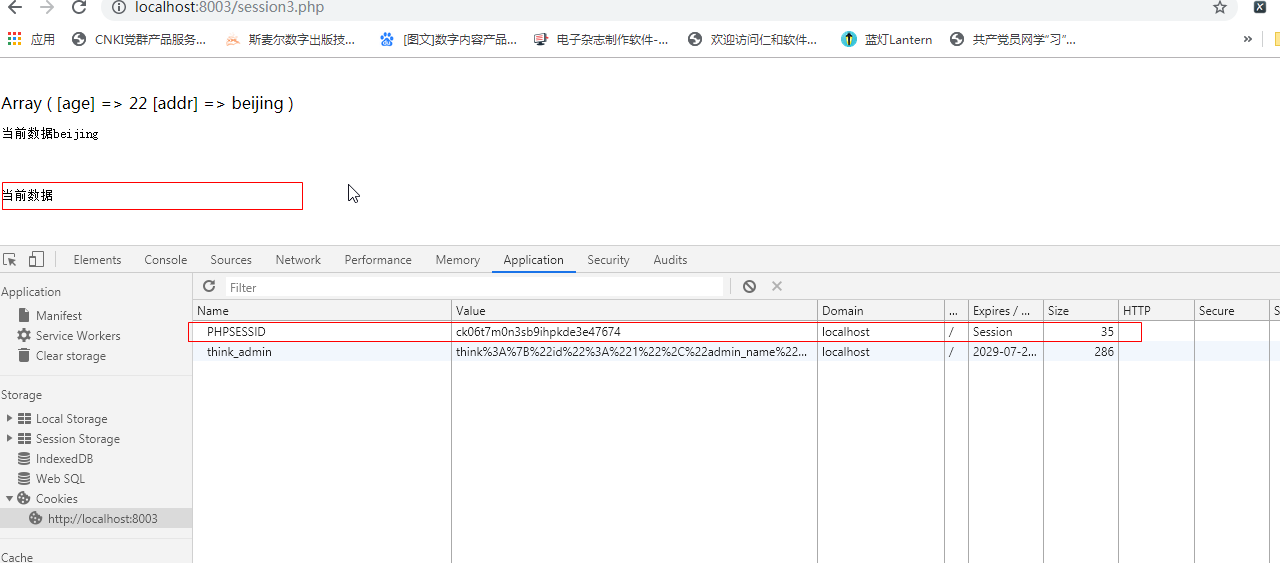
浏览器访问：



服务器session文件



销毁session后，但是不会销毁cookie中存储的session\_id，需要使用set\_cookie函数销毁



脚本：

*# 销毁session  
session\_destroy*();  
*# 销毁session，但是不会马上销毁内存中的session变量，当下次访问时，session才清空*show($\_SESSION[**'addr'**]);  
  
show(*session\_id*());

## PHP官方删除session方法

<?php

     // 初始化session.

     session\_start();

     /\*\*\* 删除所有的session变量..也可用unset($\_SESSION[xxx])逐个删除。\*\*\*\*/

     $\_SESSION = array();

     /\*\*\*删除sessin id.由于session默认是基于cookie的，所以使用setcookie删除包含session id的cookie.\*\*\*/

     if (isset($\_COOKIE[session\_name()])) {

        setcookie(session\_name(), '', time()-42000, '/');

     }

     // 最后彻底销毁session.

     session\_destroy();

?>

## session共享问题【负载均衡】

因为公司的项目之前都是部署在单台服务器上，所以在代码的编写上都没有考虑过如何在多台服务器上同时运行的情况！所以在客户考虑到高可用提出要部署多台服务器（其实也就两台啦，哈哈）的时候，考验就来啦！首先需要解决的就是sessionId的共享问题！

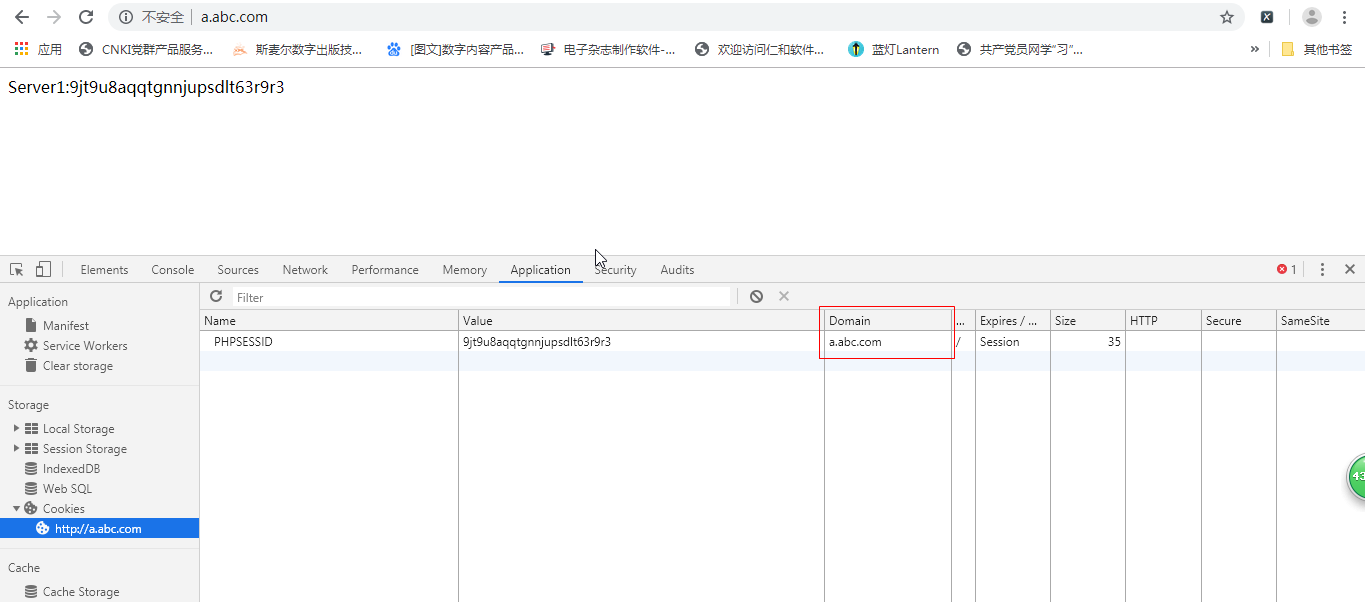
        项目部署在单台服务器运行的时候，因为客户的所有请求都是由唯一服务器来处理，sessionId保存在这台服务器上（tomcat 7）是没有问题的。但是当项目同时部署在多台服务器上时，就会出现sessionId共享问题。

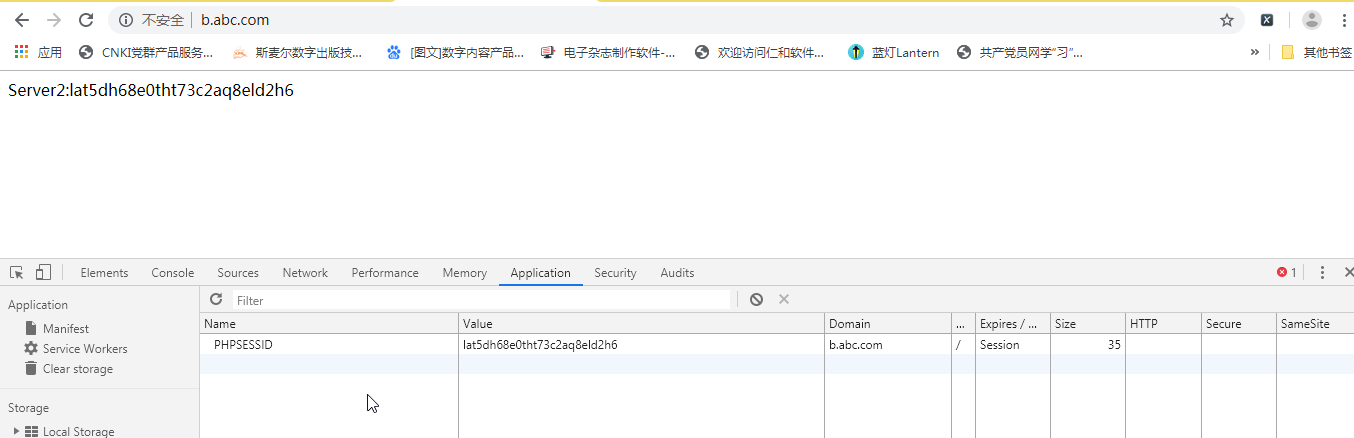
        现在有两台服务器同时运行，分别是 Server A 和 Server B，通过nginx配置了负载均衡，客户端的请求会被随机分配到两台服务器上进行处理。假设客户现在有第一次请求（登录请求）被分配到 Server A进行处理，Server A接受到请求之后会生成sessionId并且保存到内存当中，然后返回给客户（浏览器），浏览器会把sessionId保存到cookie中，第一次请求完成。如果之后每一次请求还是由Server A 来处理当然一切正常，but 一旦出现意外（比如Server A宕机），请求被分配到Server B进行处理，这时候Server B拿到客户请求的sessionId是由Server A生成的，两边对不上啊 ！ 于是客户会发现，本来还用的好好的，怎么会突然跳到登录页需要重新登录了呢！！！？？？

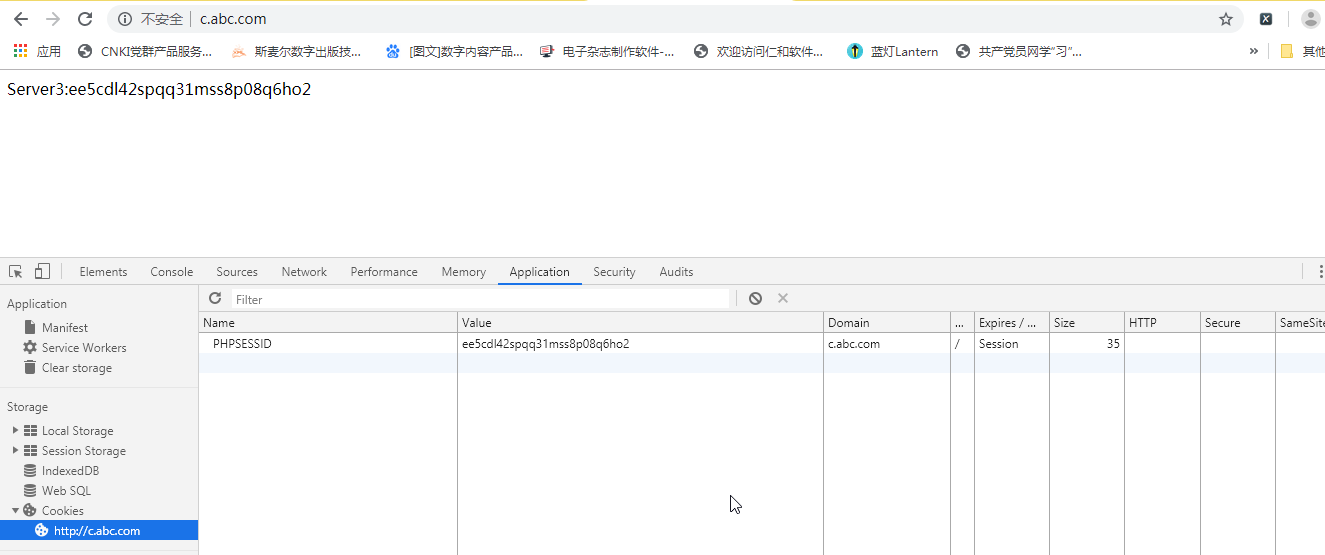
        出现问题就要思考解决方案了，既然问题的根源出在sessionId无法共享上面，那么是不是让sessionId可以在多台服务器之间共享就可以了？换个思路，即把sessionId（还有用户信息，比如user\_id）保存到数据库即可（最终选择redis,因为会比较快！），验证时不再从当前服务器获取sessionId,改从redis中获取即可！

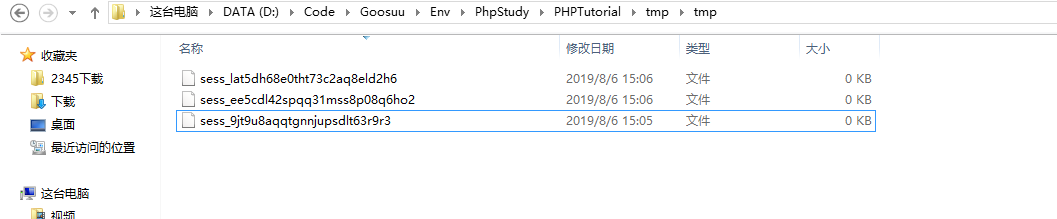
## session共享问题案例

配置三台Sever，分别为a.abc.com,b.abc.com,c.abc.com,分别访问三台服务器，访问效果如下：









脚本做更改：

**<?php** *ini\_set*(**'session.cookie\_domain'**, **'abc.com'**);  
 *session\_start*();  
 **echo "Server1:"** . *session\_id*();

