SSO单点登录讲解

概念

SSO 英文全称 Single Sign On，单点登录。

在多个应用系统中，只需要登录一次，就可以访问其他相互信任的应用系统。

比如：淘宝网（www.taobao.com），天猫网（www.tmall.com），聚划算（ju.taobao.com），飞猪网（www.fliggy.com）等，这些都是阿里巴巴集团的网站。在这些网站中，我们在其中一个网站登录了，再访问其他的网站时，就无需再进行登录，这就是 SSO 的主要用途。

好处

用户角度

用户能够做到一次登录多次使用，无需记录多套用户名和密码，省心。

系统管理员角度

管理员只需维护好一个统一的账号中心就可以了，方便。

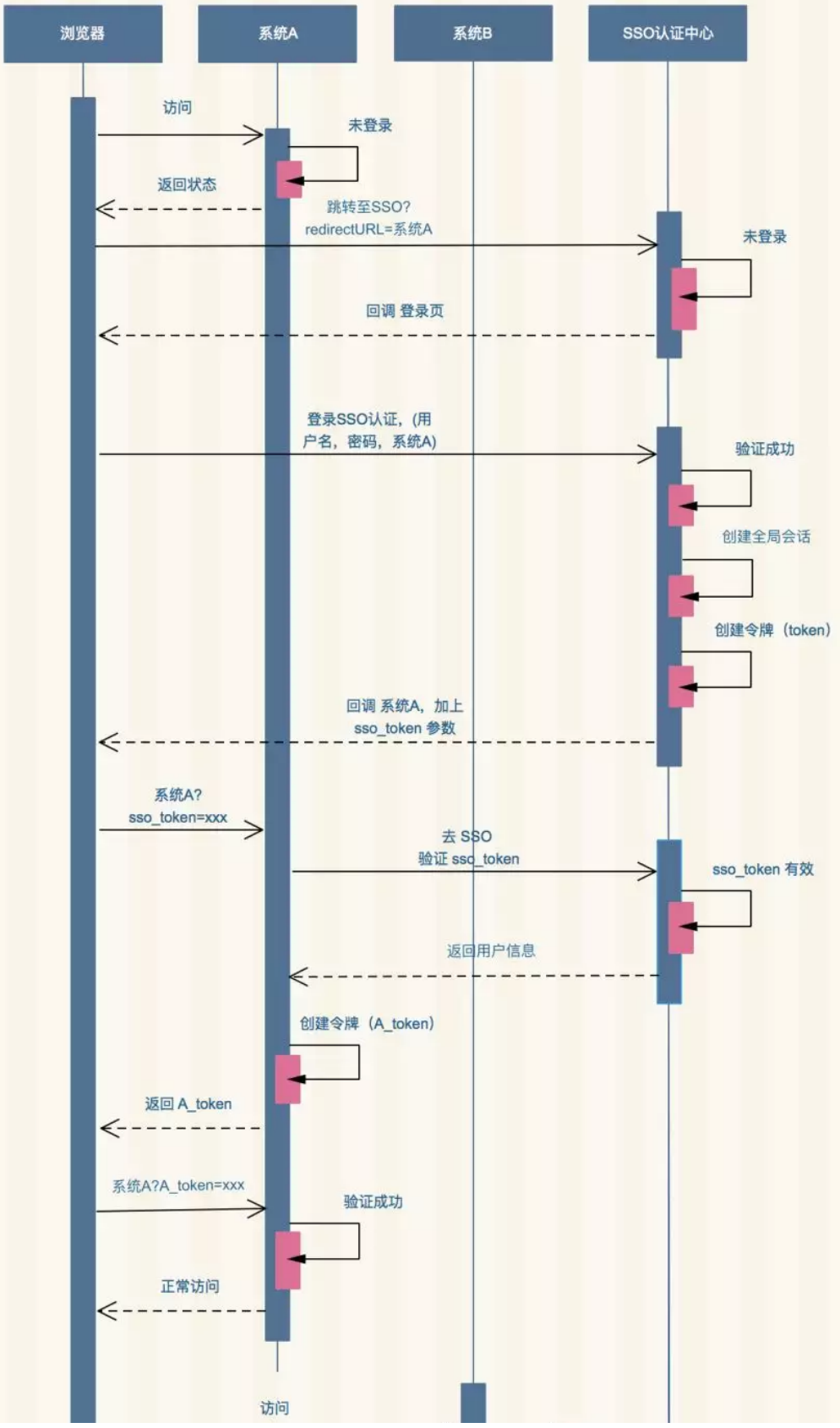
新系统开发角度

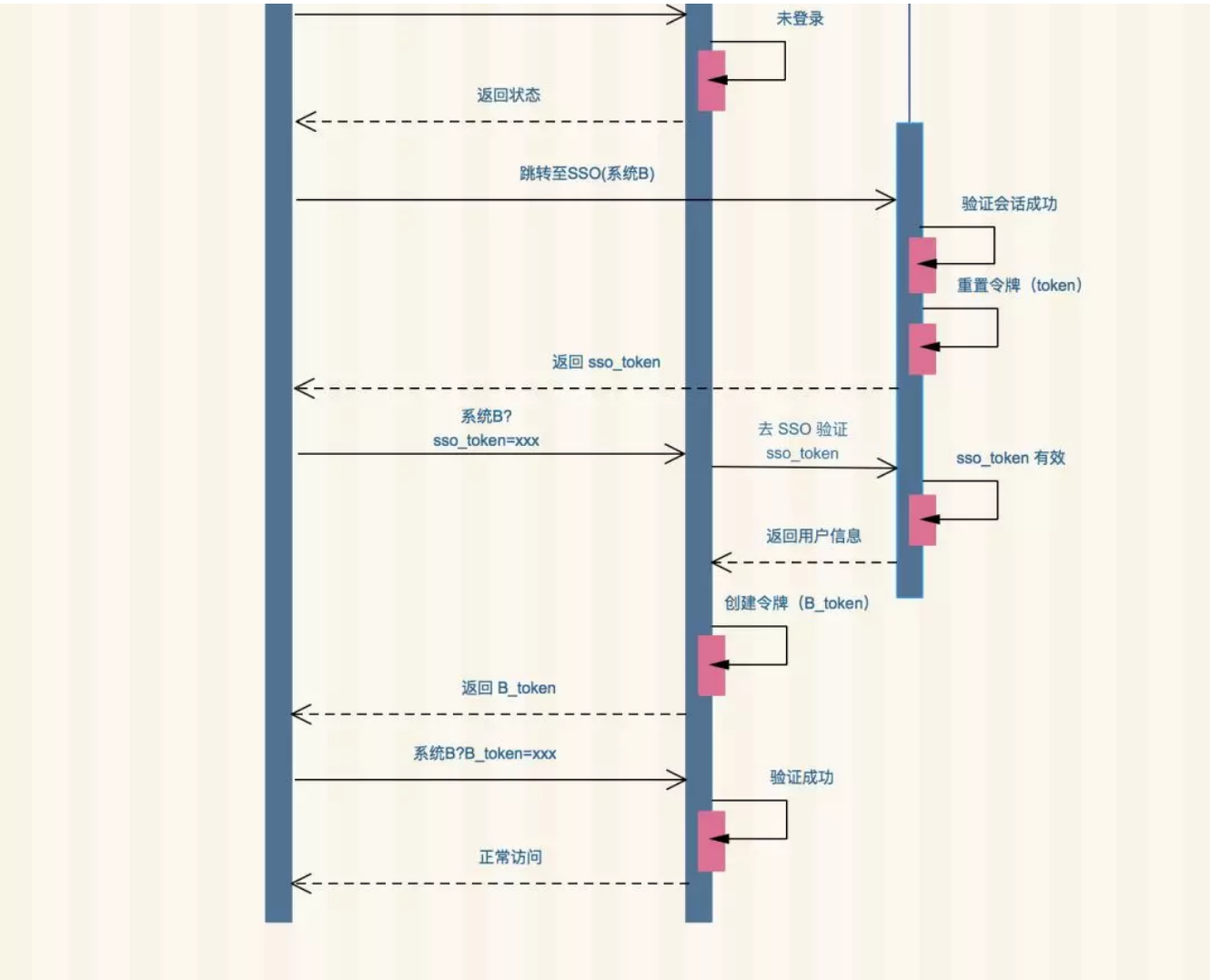
新系统开发时只需直接对接统一的账号中心即可，简化开发流程，省时。

技术实现

流程图

流程图


浏览器
系统A
系统B
SSO认证中心
访问
未登录
返回状态
跳转至SSO?
redirectURL=系统A
未登录
回调 登录页
登录SSO认证，(用
户名，密码，系统A)
验证成功
创建全局会话
创建令牌（token）
回调 系统A，加上
sso_token 参数
系统A?
sso_token=xxx
去 SSO
验证 sso_token
sso_token 有效
返回用户信息
创建令牌（A_token）
返回 A_token
系统A?A_token=xxx
验证成功
正常访问
访问



流程介绍

如果没这个介绍，看上图肯定是懵懵的。

系统A和系统B都是前后端分离的，比如前端框架用的 React / Vue / Angular，都是通过 NPM 编译后独立部署的，前后端完全通过HTTP接口的方式进行交互，也有可能前后端项目的域名都不一样。

SSO认证中心不是前后端分离的，就是前端代码和后端代码部署在一个项目中。

为什么用这两种情况呢?

其实就是为了，在流程图上出现这两种情况，这样的清楚了，后期改成任何一种就都清楚了。

试想一下:

三个系统都是前后端分离的情况，流程图应该怎么调整?

三个系统都不是前后端分离的情况，流程图应该怎么调整?

对外接口

系统A和系统B：用户退出接口。

SSO 认证中心：用户退出接口和token验证接口。

登录

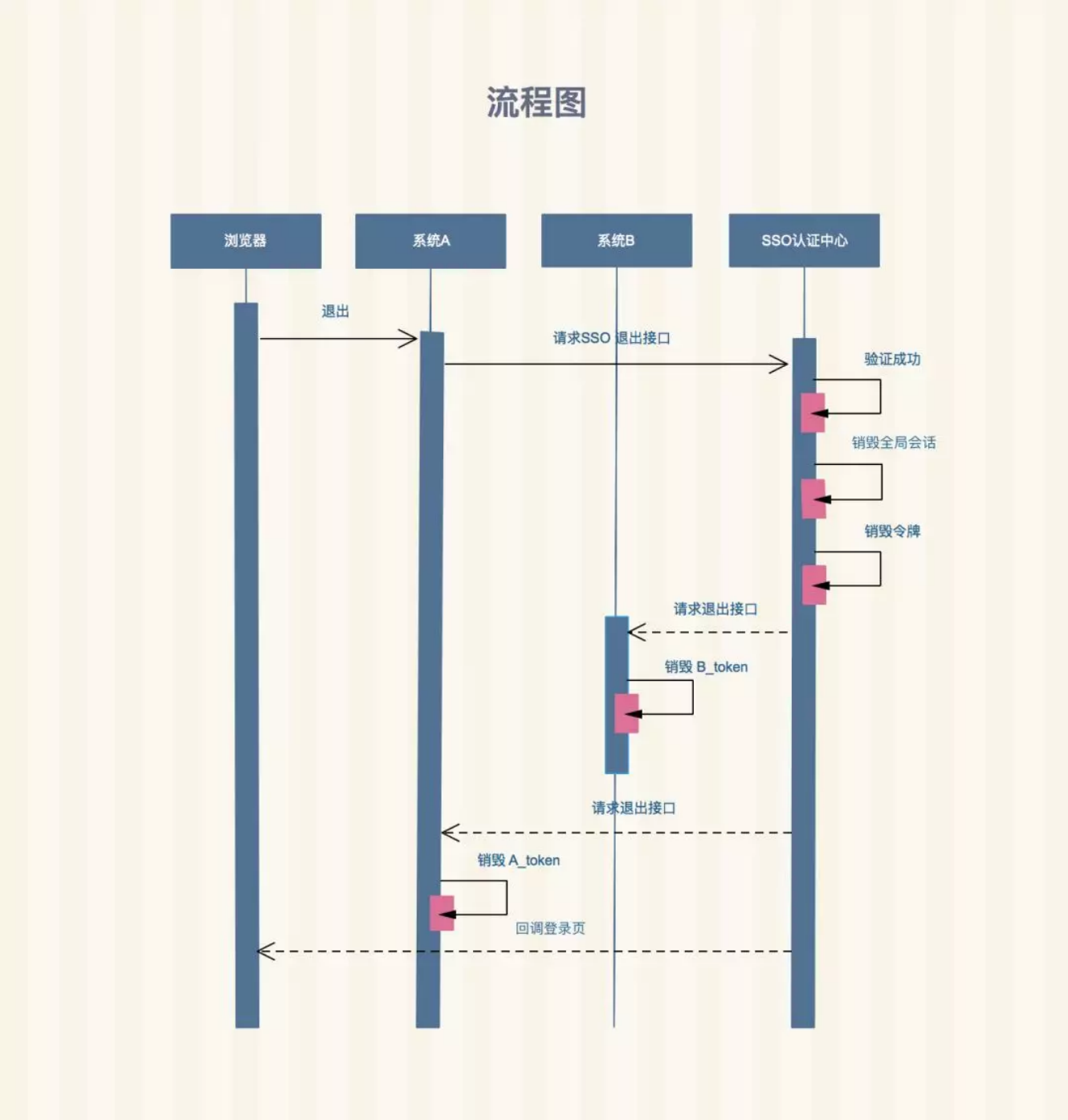
如上述流程图一致。

系统A和系统B：使用token认证登录。

SSO 认证中心：使用会话认证登录。

前后端分离项目，登录使用token进行解决，前端每次请求接口时都必须传递token参数。

退出



上图，表示的是从某一个系统退出的流程图。

退出，还可以从SSO认证中心退出，然后调取各个系统的用户退出接口。

当用户再进行操作的时候，就会跳转到SSO的登录界面。

Token 生成方式

创建全局会话可以使用session，将session存储到redis中。

令牌的生成可以使用JWT。

PHP JWT参考地址：https://github.com/lcobucci/jwt

当然还可以自定义token的生成方式。

小结

讲解了什么是SSO，以及SSO的用途与好处，同时根据流程图一步步进行梳理，基本上就可以实现了。

期间遇到任何问题，都可以关注公众号和我进行交流。

扩展

SSO与OAuth的区别

谈到SSO很多人就想到OAuth，也有谈到OAuth想到SSO的，在这里我简单的说一下区别。

通俗的解释，SSO是处理一个公司内的不同应用系统之间的登录问题，比如阿里巴巴旗下有很多应用系统，我们只需要登录一个系统就可以实现不同系统之间的跳转。

OAuth是不同公司遵循的一种授权方案，也是一种授权协议，通常都是由大公司提供，比如腾讯，微博。我们常用的QQ登录，微博登录等，使用OAuth的好处是可以使用其他第三方账号进行登录系统，减少了因用户懒，不愿注册而导致用户流失的风险。

现在一些支付业务也用OAuth，比如微信支付，支付宝支付。

还有一些开放平台也用OAuth，比如百度开放平台，腾讯开放平台。

SSO与RBAC的关系

如果企业有多个管理系统，现由原来的每个系统都有一个登录，调整为统一登录认证。

那么每个管理系统都有权限控制，吸取统一登录认证的经验，我们也可以做一套统一的RBAC权限认证。