GVP — Gitee最有价值开源项目

符节科技 JustAuthPlus

JAP 产品技术白皮书

开源的统一登录认证标准组件

V 1.0.1

北京符节科技有限公司

2021 年 10 月

© 2020-2040 符节科技 版权所有

编写/修订成员

张亚东、沈扬凯、JAP 开发团队

编写/修订时间

2021 年 10 月

完成时间

2021 年 10 月

项目地址

https://gitee.com/fujieid/jap
https://github.com/fujieid/jap
https://codechina.csdn.net/fujieid/jap

符节科技 JustAuthPlus

© 2020-2040 符节科技 版权所有

目录

符节科技 JustAuthPlus

© 2020-2040 符节科技 版权所有

前言

本白皮书内涉及的专业性的专有名词，在“附录一”中会有详细解释，协助你更深的了解本篇“白皮书”。

一、背景概述

现如今，几乎每个 WEB 应用，都要开发并支持各种登录方式，比如账号密码登录、OAuth 登录、OIDC 登录、SAML 登录、MFA 登录、单点登录等，而这一部分功能是高度相似或重复的。如 OAuth 和 SAML 这般需求，则需要研发人员花费大量的时间和精力去研究、测试。

1.1 传统做法

为每个项目都实现一套登录相关的功能代码，如图

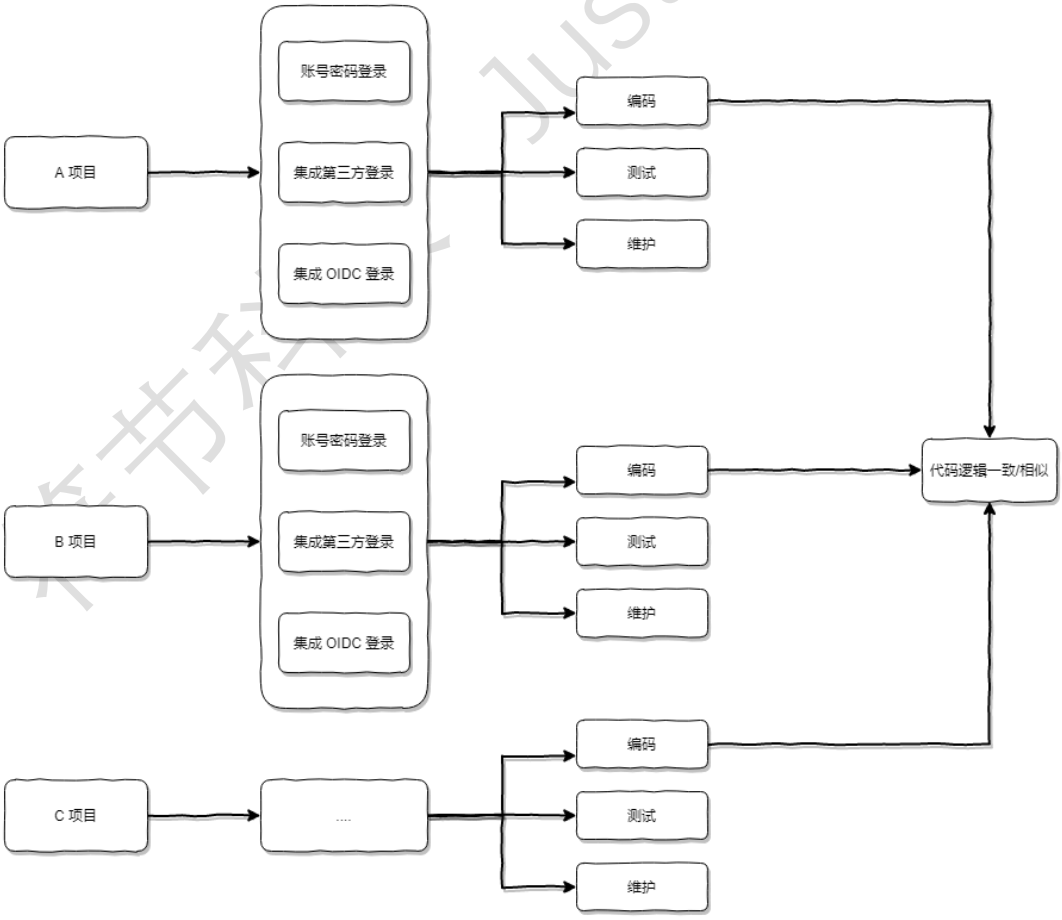

图 1.1 传统登录模块开发示意图

© 2020-2040 符节科技 版权所有

这势必会造成以下问题：

1. 给研发人员带来大量的重复劳动
2. 增加代码的维护成本
3. 影响到研发产出效率
4. 耗费企业的人力成本

1.2 当前需求

基于以上问题，开发者切实需要一种能够完美支持多种登录场景并且开箱易用的产品或者解决方案。

一、产品介绍

JustAuthPlus（以下简称"**JAP**"）是一款开源的**登录认证中间件**，基于模块化设计，为所有需要登录认证的 WEB 应用提供一套标准的技术解决方案，开发者可以基于 JAP 适配绝大多数~~的~~ WEB 系统（自有系统、联邦协议）。

JAP 支持任何基于标准 OAuth 协议、OIDC 协议的平台/应用，与开发者实际的业务逻辑完美解耦，对外提供统一的接入标准，支持多语言版本等。

开发者可无缝对接绝大多数第三方应用或者自有 WEB 系统，快速实现登录模块，提高开发效率，减少代码维护成本。

简单来说，JAP 是基于 JustAuth 研发的升级产品或者称为专业级产品，JustAuth 解决的是第三方登录的问题，JAP 解决的是整个登录相关的问题。

2.1 创作目的

为所有需要登录认证的应用提供一套标准的解决方案，方便开发者无缝对接其他第三方应用或者自有的 WEB 应用。

2.2 主要用途

方便开发者无缝对接绝大多数第三方应用或者自有的 WEB 应用，快速实现登录模块，提高开发效率，减少代码维护成本。我们更倾向于让开发者可以基于 JAP 实现自己的 IDaaS/IAM 系统。**Build your own IDaaS/IAM**。

© 2020-2040 符节科技 版权所有

2.3 主要功能

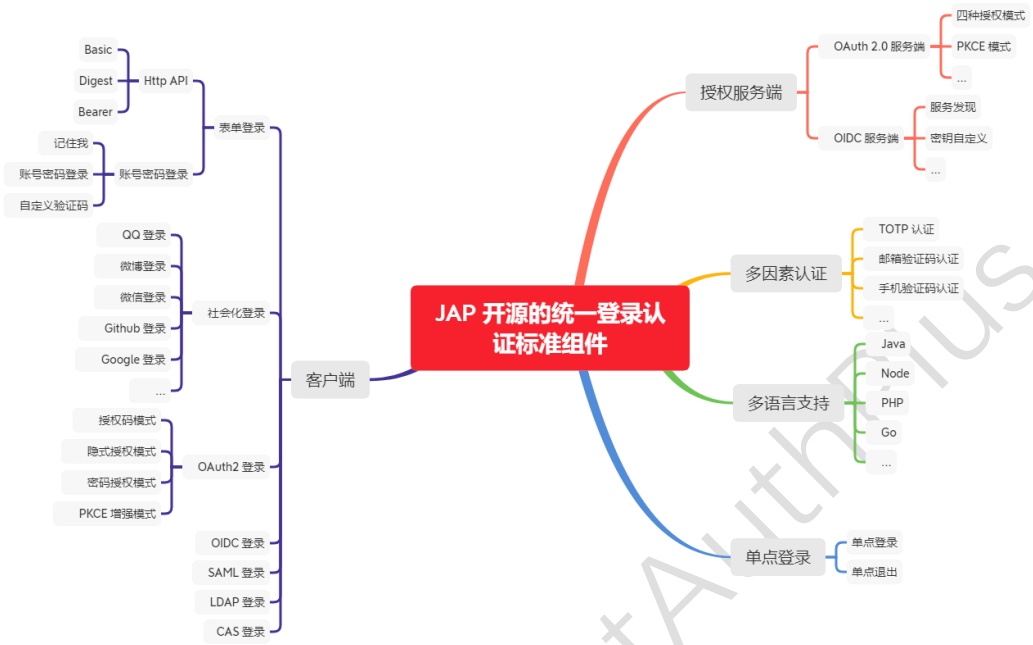

图 2.3 JAP 功能图示

2.4 产品优势

JAP 的优势在于易用性、全面性、模块化、标准化和通用性：

- **易用性：**JAP 的 API 沿袭 JustAuth 的简单性，做到了开箱即用的程度。JAP 高度抽象各种登录场景，提供了多套简单使用的 API，极大程度的降低了开发者的学习成本和使用成本，部分 API 如下：

```
SimpleStrategy simpleStrategy = new SimpleStrategy(japUserService, new JapConfig());
simpleStrategy.authenticate(new SimpleConfig(), requestAdapter, responseAdapter)
```

图 2.4-1 账号密码登录

© 2020-2040 符节科技 版权所有

```
SocialStrategy socialStrategy = new SocialStrategy(japUserService, new JapConfig());
socialStrategy.authenticate(new SocialConfig()
                .setPlatform("gitee")
                .setState(UuidUtils.getUUID())
                .setJustAuthConfig(AuthConfig.builder()
                        .clientId("xxxx")
                        .clientSecret("xxxx")
                        .redirectUri("http://xxx.xx")
                        .build()), requestAdapter, responseAdapter)
```

图 2.4-2 社会化账号登录

```
Oauth2Strategy oauth2Strategy = new Oauth2Strategy(japUserService, new JapConfig());
oauth2Strategy.authenticate(new OAuthConfig().setPlatform("gitlab")
                .setState(UuidUtils.getUUID())
                .setClientId("xx")
                .setClientSecret("xx")
                .setCallbackUrl("http://xxx.xx")
                .setAuthorizationUrl("https://xxx.xx/oauth/authorize")
                .setTokenUrl("https://xxx.xx/oauth/token")
                .setUserinfoUrl("https://xxx.xx/api/v4/user")
                .setScopes(new String[]{"profile"})
                .setResponseType(Oauth2ResponseType.CODE)
                .setGrantType(Oauth2GrantType.AUTHORIZATION_CODE)
                .setUserInfoEndpointMethodType(Oauth2EndpointMethodType.GET),
        requestAdapter, responseAdapter)
```

图 2.4-3 OAuth 2.0 协议的账号登录

```
OidcStrategy oidcStrategy = new OidcStrategy(japUserService, new JapConfig());
oidcStrategy.authenticate(new OidcConfig()
                .setPlatform("xxx")
                .setIssuer("xx")
                .setState(UuidUtils.getUUID())
                .setClientId("xx")
                .setClientSecret("xx")
                .setCallbackUrl("http://xxx.xx")
                .setScopes(new String[]{"read", "write"})
                .setResponseType(Oauth2ResponseType.CODE)
                .setGrantType(Oauth2GrantType.AUTHORIZATION_CODE), requestAdapter, responseAdapter)
```

图 2.4-4 OpenID Connect 协议的账号登录

© 2020-2040 符节科技 版权所有

```
JapStrategy ldapStrategy = new LdapStrategy(japUserService, new JapConfig());
ldapStrategy.authenticate(new LdapConfig()
                        .setUrl("ldap://localhost:389")
                        .setBindDn("cn=admin,dc=test,dc=com")
                        .setCredentials("123456")
                        .setBaseDn("dc=test,dc=com")
                        .setFilters("(&(objectClass=inetOrgPerson)(uid=%s))")
                        .setTrustStore("")
                        .setTrustStorePassword(""),
                new JakartaRequestAdapter(request),
                new JakartaResponseAdapter(response))
```

图 2.4-5 LDAP 的账号登录

```
HttpApiStrategy httpApiStrategy = new HttpApiStrategy(new JapHttpApiUserServiceImpl(), new JapConfig());
httpApiStrategy.authenticate(new HttpApiConfig()
                    .setHttpMethod(HttpMethodEnum.GET);
                    .setAuthInfoField(AuthInfoFieldEnum.BODY);
                    .setAuthSchema(AuthSchemaEnum.BASIC);
                    .setLoginUrl("http://httpbin.org/basic-auth/foo/bar"),
                new JakartaRequestAdapter(request),
                new JakartaResponseAdapter(response))
```

图 2.4-6 Http API 的账号登录

以实现 OAuth 登录为例，未使用 JAP 时的研发流程：

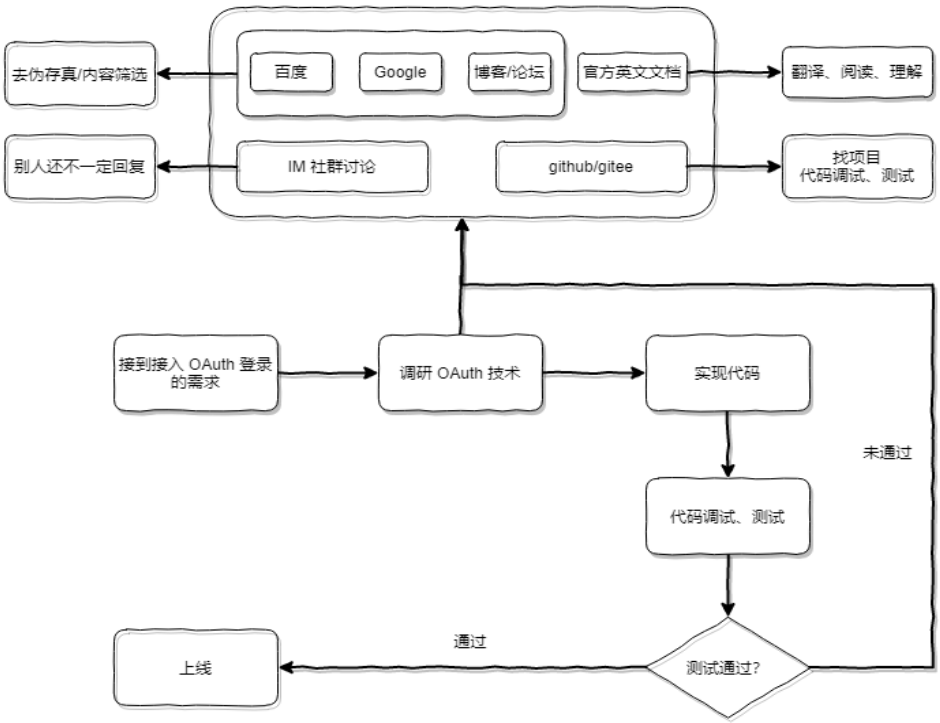

图 2.4-5 未使用 JAP 时开发 OAuth 登录的流程

从上图可以看出，开发人员将会把大多数时间用到“调研 OAuth 技术”阶段，而开发者对于 OAuth 实现协议一旦理解不够深入或者有偏差，在编码实现阶段，就一定会遇到问题，进而需要再次调研相关知识，以此往复直到问题解决。而使用 JAP 之后的流程就简单多了：

© 2020-2040 符节科技 版权所有

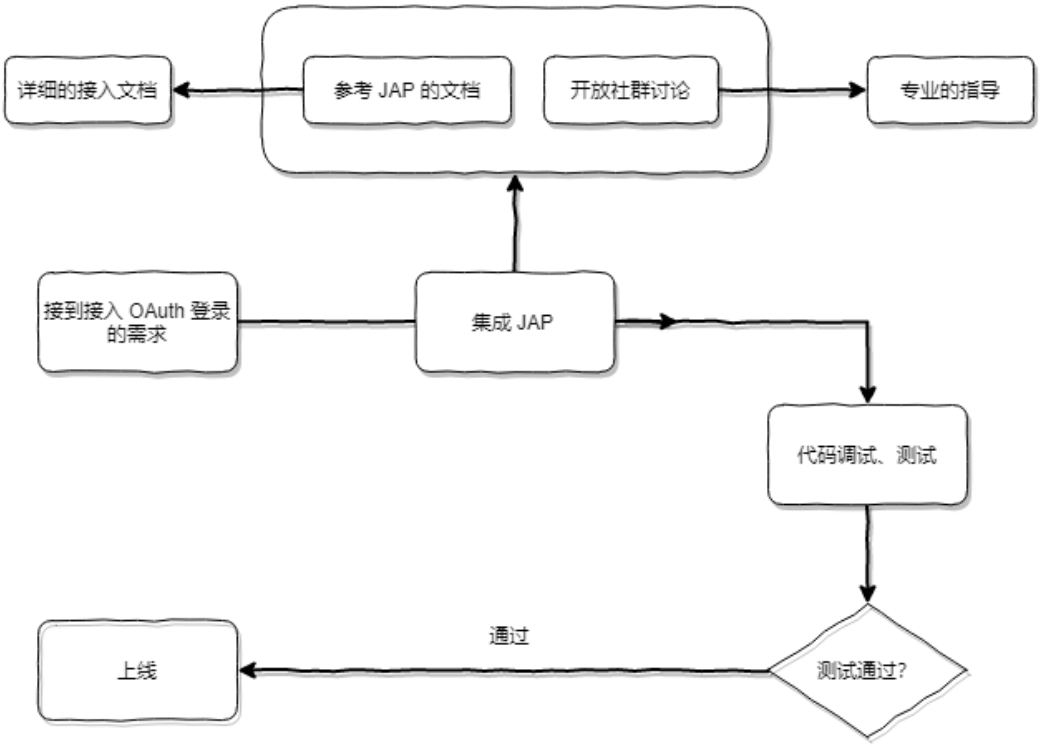

图 2.4-6 使用 JAP 后开发 OAuth 登录的流程

开发者不再需要关注 OAuth 的具体实现逻辑，只需要关注 OAuth 应用的创建即可。通过 JAP 简单的 API 就可以实现完整且复杂的 OAuth 授权流程。

- **全面性：**JAP 全量适配 JustAuth 支持的第三方平台，实现第三方登录。同时也支持所有基于标准 OAuth2.0 协议或者 OIDC 协议或者 SAML 协议的应用、系统，同时 JAP 还提供不同语言版本的项目 SDK，适配多种研发场景；

图 2.4-7 支持的第三方登录平台

- **模块化：**JAP 基于模块化设计开发，针对每一种登录场景，比如账号密码、OAuth、OIDC 等，都单独提供了独有的模块化解决方案；

- **标准化：**JAP 和业务完全解耦，将登录认证相关的逻辑抽象出一套标准的技术解决方案，针对每一种业务场景，比如用户登录、验证密码、创建并绑定第三方系统的账号等，都提供了一套标准的策略或者接口，开发者可以基于 JAP，灵活并方便的完成相关业务逻辑的开发和适配；

© 2020-2040 符节科技 版权所有

```java
public interface JapUserService {
    JapUser getById(String userId);
    JapUser getByName(String username);
    boolean validPassword(String password, JapUser user);
    JapUser getByPlatformAndUid(String platform, String uid);
    JapUser createAndGetSocialUser(Object userInfo);
    boolean bindSocialUser(JapUser japUser, String bindUserId);
    JapUser createAndGetOauth2User(String platform, Map<String, Object> userInfo, Object tokenInfo);
    void saveHttpAuthedJapUser(JapUser japUser);
    JapUser createAndGetLdapUser(Object userInfo);
}
```

图 2.4-9 JAP 提供的标准接口

- **通用性：**JAP 不仅可以用到第三方登录、OAuth 授权、OIDC 认证等业务场景，还能适配开发者现有的业务系统的普通账号密码的登录场景，基本将所有登录相关的业务场景都已经涵盖。针对 WEB 应用，JAP 将提供满足各种不同登录场景的解决方案（和开发语言无关）。

2.5 技术特色

JAP 基于模块化分层和行业标准协议开发，依托于开源社区，这些给我们带来如下优势：

1. 广泛的业界支持：如 RFC6749、OIDC 等协议都是相关领域流行和默认的标准。
2. 提高技术安全性：标准的协议已经通过大量的业务场景验证，保证了技术方案的成熟性和安全性。
3. 提高产品维护性：基于开源的场景确保了各种社区的活跃度，可以更好的解决产品维护过程中遇到的问题；基于模块化的分层结构，便于快速聚焦问题、快速响应。
4. 提高研发效率：基于开源的场景会有大量的开发人员提供大量个性化的解决方案，能更快速的找到满足需求的各种解决方案。

三、项目架构

为了防止开发者在一开始使用 JAP 时感到复杂，在此对 JAP 的项目架构进行介绍说明，降低开发者的学习成本。JAP 基于模块化设计开发，每一个模块都对应一类具体的业务场景，可以更加有针对性地解决开发者的问题。具体的模块说明，如下：

© 2020-2040 符节科技 版权所有

3.1 模块说明

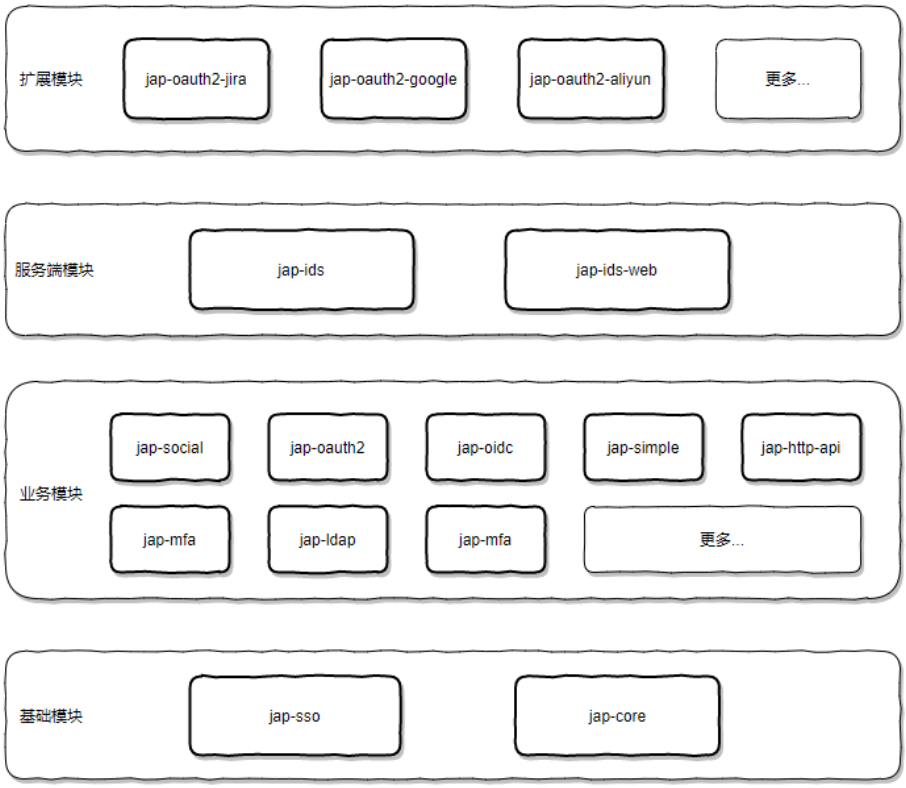

图 3.1-1 JAP 模块设计

- **jap-sso**：提供单点登录的实现逻辑。JAP 对单点登录做了一定的优化，开发者甚至只需要一行代码就可开启单点登录功能；
- **jap-core**：JAP 核心模块，对 JAP 项目下其他模块（或者周边项目）提供标准的接口规范；
- **jap-simple**：适用于普通账户密码登录，同时提供了“记住我”的功能；
- **jap-social**：JAP 集成 JustAuth ，完美适配 JustAuth 支持的所有第三方平台的授权登录；
- **jap-oauth2**：JAP 基于 OAuth2.0 协议（RFC6749），实现的 OAuth2.0 客户端，可以基于此对接任何支持 OAuth2.0 标准协议的平台，并且基于 JAP 的极简设计思想，极大的降低了 OAuth2 接入的复杂性；
- **jap-oidc**：JAP 基于 OIDC 协议（OpenID-Connect-Core-1.0），实现的 OIDC 客户端，它可以对接任何支持 OIDC 标准协议的平台，并且基于 JAP 的极简设计思想，极大的降低了 OIDC 接入的复杂性；
- **jap-mfa**：封装实现 MFA 多因素认证，目前内置有 TOTP（Time-based One-time Password） 认证；
- **jap-ids**：JAP 基于 OAuth2.0 协议（RFC 6749）和 OIDC 协议（OpenID-Connect-Core-1.0），实现的授权认证服务端（Identity Server，简称“IDS”）；
- **jap-ids-web**：配合 jap-ids 模块使用，提供权限过滤器；
- **jap-ldap**：封装 LDAP 的基本操作，实现 LDAP 用户的登录认证；

© 2020-2040 符节科技 版权所有

- **jap-http-api**: 实现 HTTP API 方式的登录认证，支持 Basic、Digest 和 Bearer 等方式；
- 更多...

3.2 模块间的依赖关系

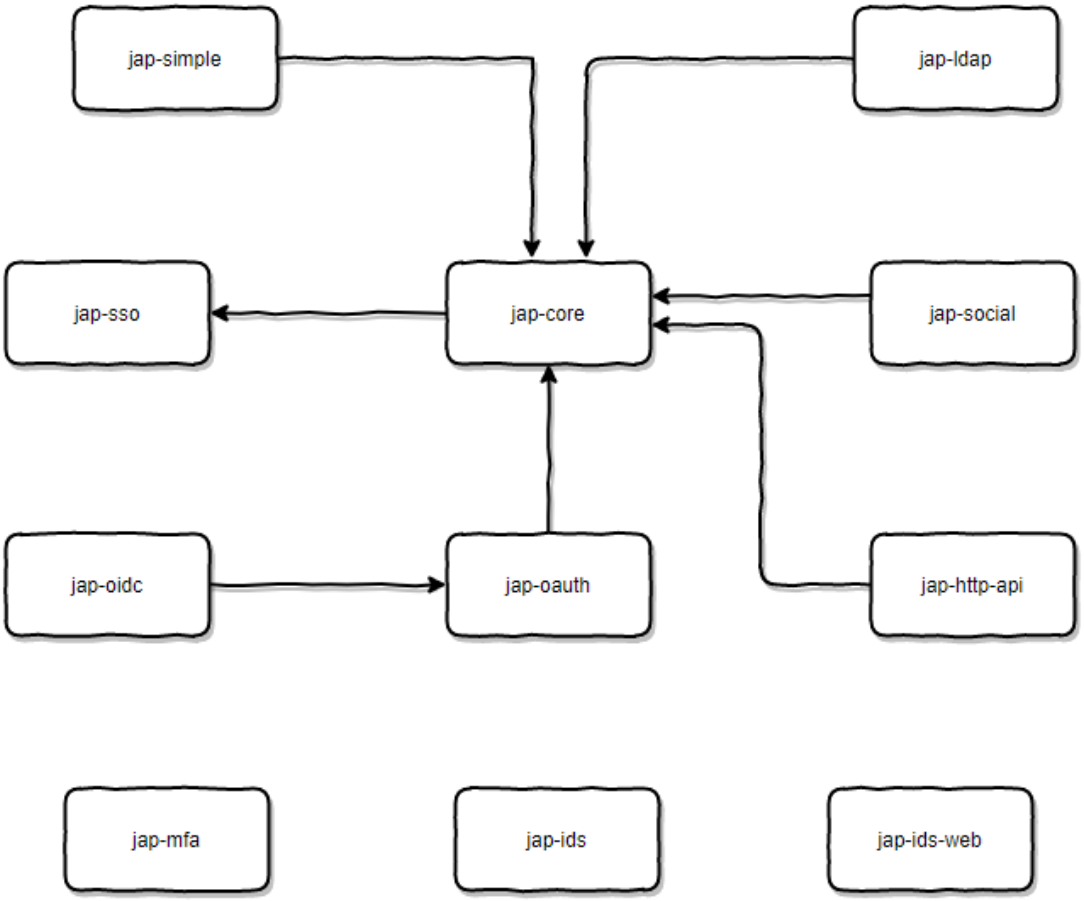

图 3.2 JAP 各模块间的依赖关系（箭头表示依赖）

© 2020-2040 符节科技 版权所有

3.3 业务流程图

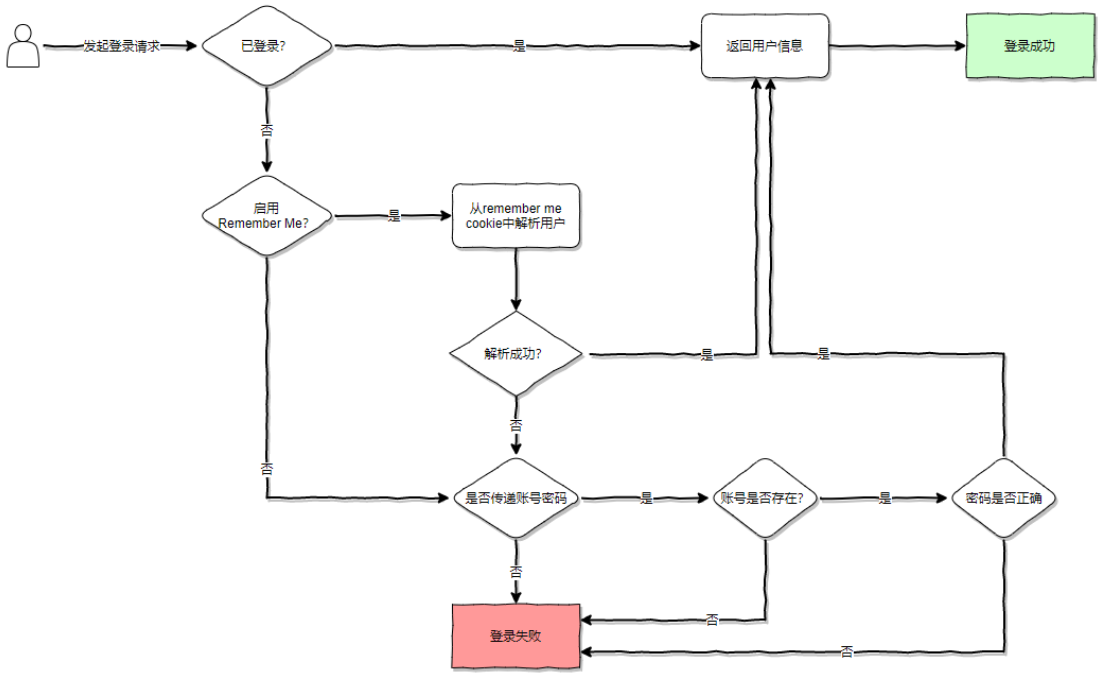

图 3.3-1 jap-simple 模块的业务流程图

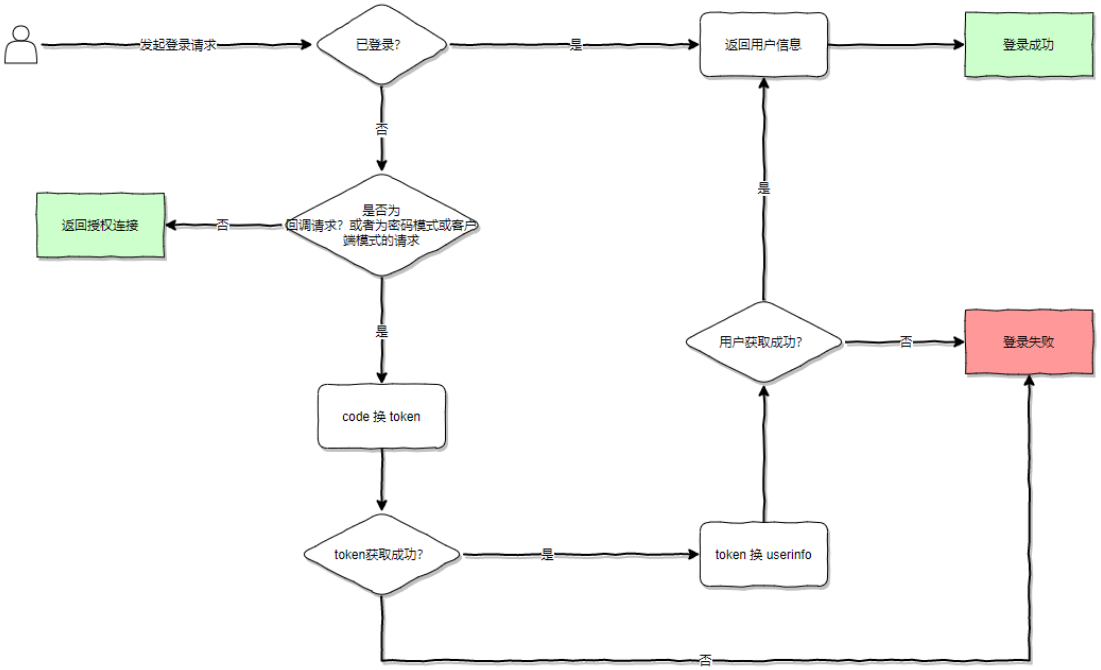

图 3.3-2 jap-oauth2 模块的业务流程图

© 2020-2040 符节科技 版权所有

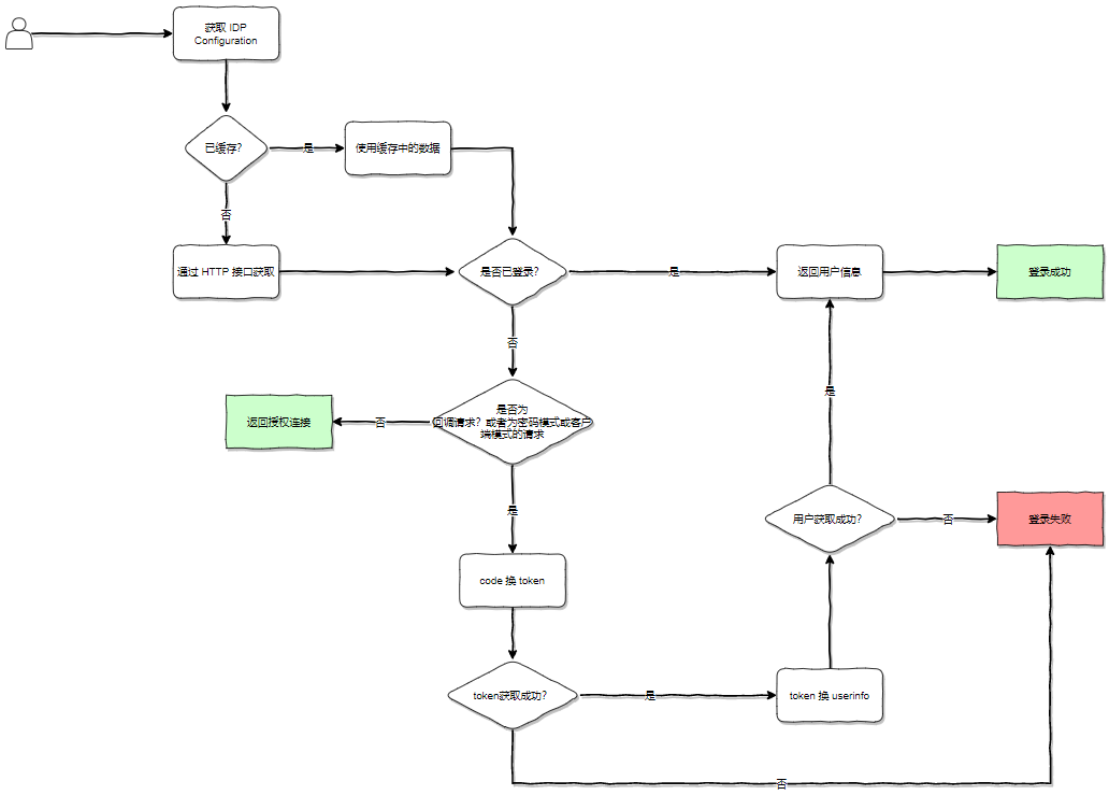

图 3.3-3 jap-oidc 模块的业务流程图

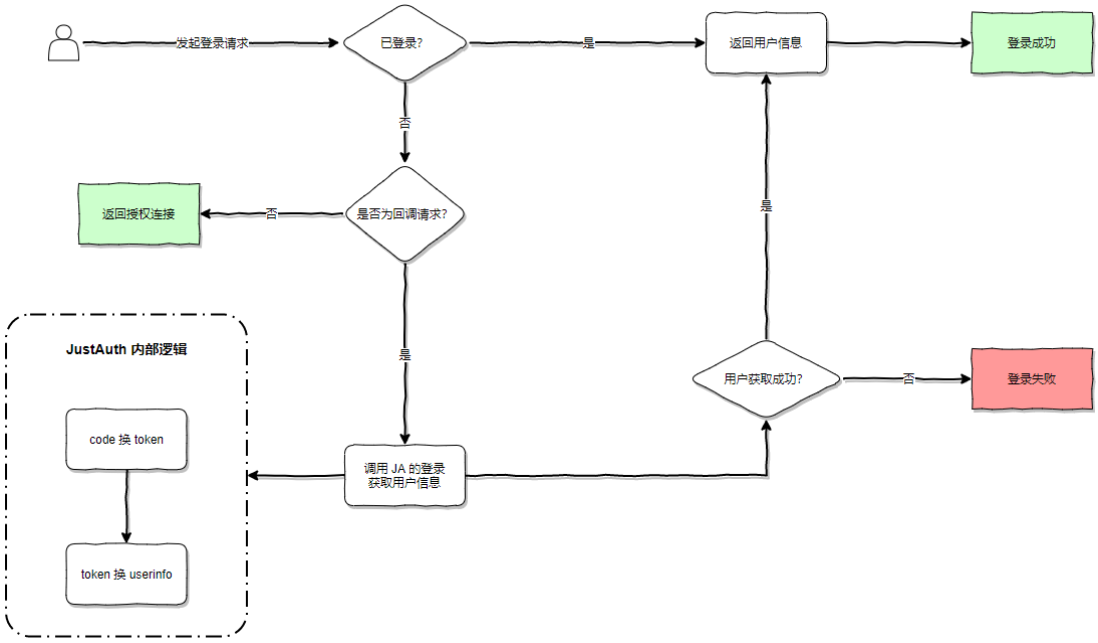

图 3.3-4 jap-social 模块的"登录"业务流程图

© 2020-2040 符节科技 版权所有

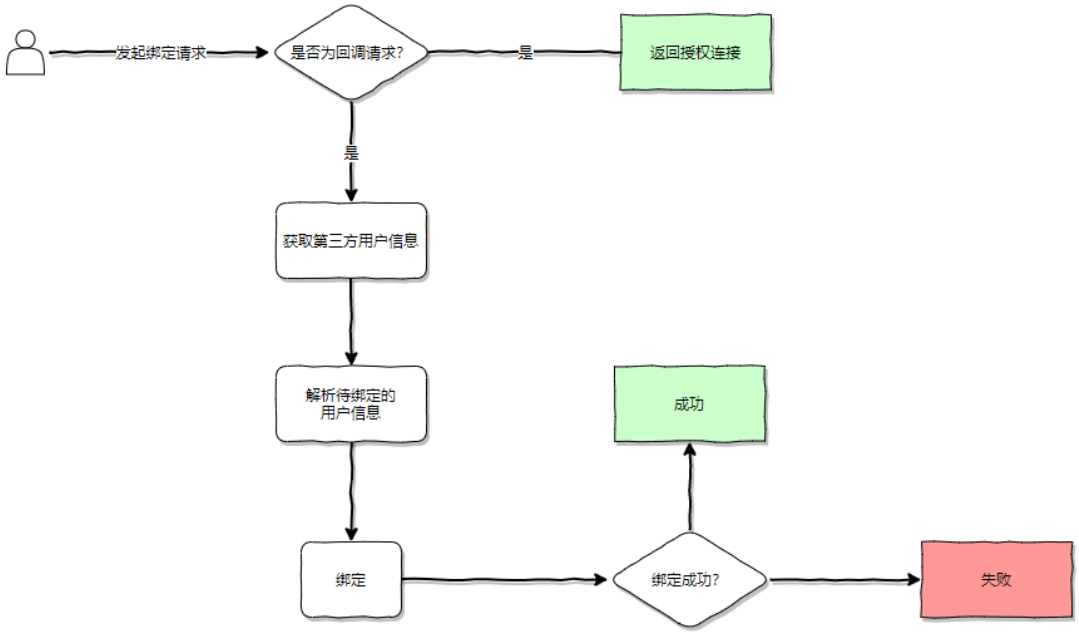

图 3.3-5 jap-social 模块的”绑定"业务流程图

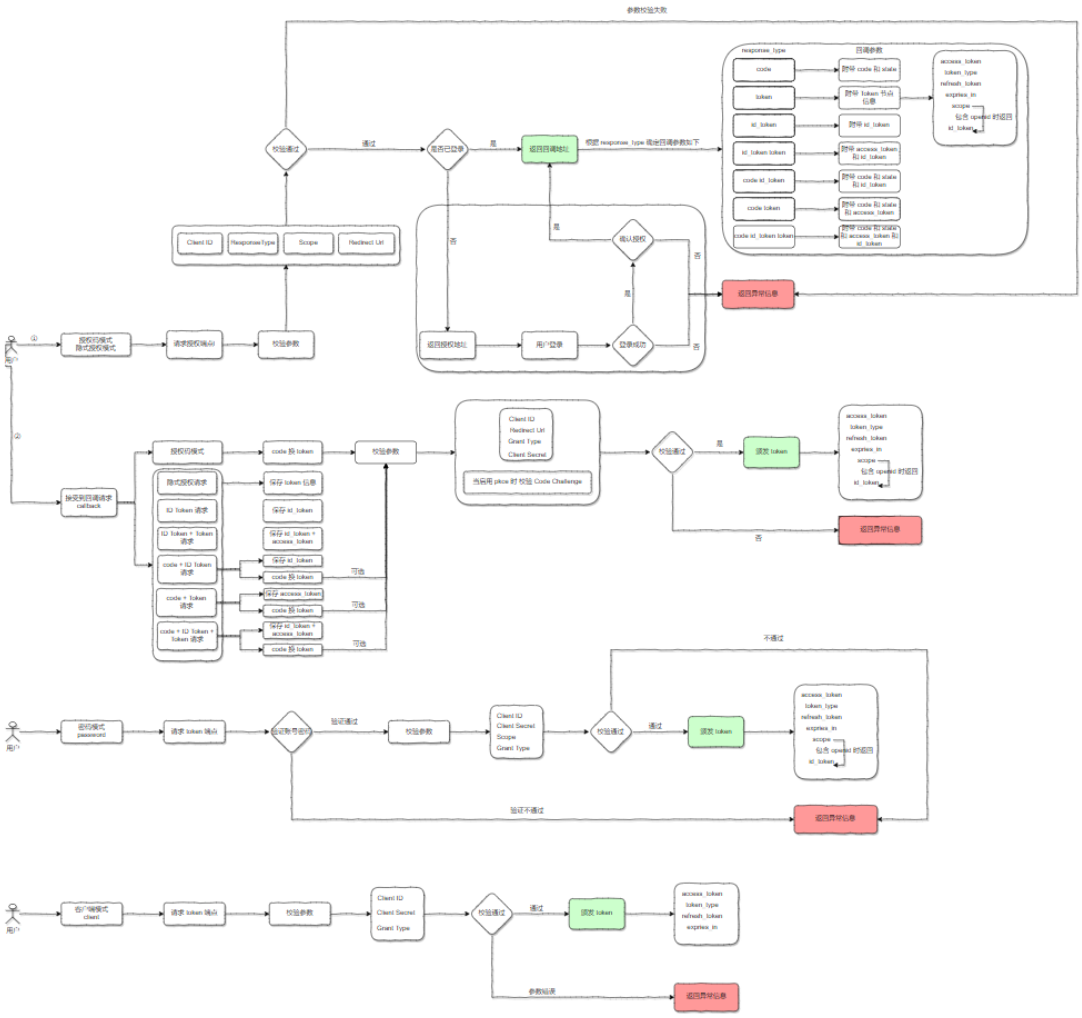

图 3.3-6 jap-ids 模块的业务流程图

© 2020-2040 符节科技 版权所有

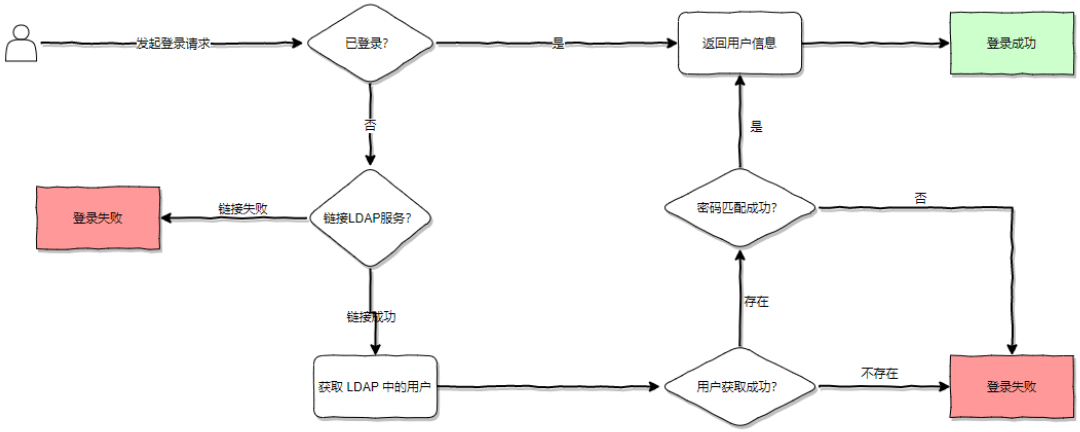

图 3.3-7 jap-ldap 模块的业务流程图

四、JAP 适用的业务场景

JAP 适用于所有需要登录功能的业务场景。比如：

- **标准规范**：新项目立项，你们需要研发一套包含登录、认证的系统，并且需要一套标准的、灵活的、功能全面的登录认证方案。
- **需求灵活**：现有登录模块为自研，但是新一轮的技术规划中，你们想将登录认证模块重构，以更加灵活的架构适应后面的新需求，比如：集成 MFA 登录、集成 OAuth 登录、SAML 登录等。
- **力求省事**：你们的项目太多（或者是开发语言较多，比如：Java、Python、Node 等），每个项目都需要登录认证模块，想解决这种重复劳动的问题，使研发人员有更多的时间和精力投入到业务开发中，提高研发产能和研发效率。

五、未来规划

JAP 的功能远不止于此，后面我们会对 JAP 持续不断的迭代、改进，后面会逐步实现如下功能：实现 LDAP 授权登录。

1. 实现 SAML 授权登录。
2. 实现 Form-based 授权登录。
3. 实现跨域单点登录。
4. 开发并发布不同语言版本的 SDK，如：NodeJS、Python、PHP、GO 等。
5. 基于 JAP 提供一整套统一身份认证的解决方案。
6. 基于 RBAC、ABAC、PBAC 提供权限管理。
7. 更多...

© 2020-2040 符节科技 版权所有

欢迎各位开发者共同参与，一起将 JAP 打造成登录认证领域内人人都在用、人人都可用的开源技术组件。

六、总结

JAP 沿袭了 JustAuth 的简单、全面和易用的设计理念，为登录认证需求提供了一套通用且标准的技术解决方案，能够适配如账号密码登录、OAuth 登录、OIDC 登录、单点登录等多种业务场景，基于模块化的架构设计，使得 JAP 可以灵活的适配各种不同的登录相关的业务场景，为登录认证功能的标准化、通用化、易用化助力。

JAP 将在后续的版本中，围绕登录认证这一核心业务场景，完善相关技术方案、提供更多的测试用例和功能，使其越来越好用。

JAP 以开源的方式，受惠于开源社区，赋能于开发者。使之成为开发者生态内必不可少的"基础设施"，以期形成登录场景下的标准解决方案。我们更倾向于让开发者可以基于 JAP 实现自己的 IDaaS/IAM 系统。**Build your own IDaaS/IAM。**

七、参考资料

1. JustAuth · https://github.com/justauth/JustAuth
2. OpenID Connect Core 1.0 incorporating errata set 1 · https://openid.net/specs/openid-connect-core-1_0.html
3. The OAuth 2.0 Authorization Framework · https://tools.ietf.org/html/rfc6749
4. OAuth · https://en.wikipedia.org/wiki/OAuth
5. SAML Wiki · https://wiki.oasis-open.org/security/FrontPage
6. Multi-factor authentication · https://en.wikipedia.org/wiki/Multi-factor_authentication
7. One-time password · https://en.wikipedia.org/wiki/One-time_password
8. Time-based One-Time Password · https://en.wikipedia.org/wiki/Time-based_One-Time_Password
9. TOTP: Time-Based One-Time Password Algorithm · https://tools.ietf.org/html/rfc6238
10. Identity management · https://en.wikipedia.org/wiki/Identity_management
11. Lightweight Directory Access Protocol · https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

© 2020-2040 符节科技 版权所有

八、联系我们

微信交流群

扫码进入 JAP 社群，备注“JAP”

公众号

扫码关注公众号

开发者文档

https://justauth.plus

开发者社区

https://discuss.justauth.plus

开源项目

https://gitee.com/fujieid/jap
https://github.com/fujieid/jap
https://codechina.csdn.net/fujieid/jap

邮箱

dev@fujieid.com

符节科技 JustAuthPlus

© 2020-2040 符节科技 版权所有

附录一

- **OAuth**：OAuth 是一套标准化的授权协议，通常用于互联网用户授予网站或应用程序访问其在其他网站上的信息的权限，但不提供密码。OAuth2.0 为 web 应用程序、桌面应用程序、移动电话和智能设备提供了特定的授权流，安全性方面要比 OAuth1.0 更优秀。
- **OIDC**：OpenID Connect 是在 OAuth2.0 协议之上的一个简单的身份层。它允许客户端基于授权服务器执行的身份验证来验证最终用户的身份，并以可互操作和类似于 REST 的方式获取有关最终用户的基本配置文件信息。OpenID Connect 允许各种各样的客户端，包括基于 Web 的客户端、移动客户端和 JavaScript 客户端。
- **IDS**：Identity Server，身份服务
- **SAML**：SAML（安全断言标记语言（英语：Security Assertion Markup Language，简称 SAML，发音 sam-el））是用于 Web 浏览器单点登录的基于 XML 的标准，由 OASIS 安全服务技术委员会定义，用于在不同的安全域(security domain)之间用户身份验证和授权数据交换。SAML 2.0 可以实现基于网络跨域的单点登录（SSO），以便于减少向一个用户分发多个身份验证令牌的管理开销。SAML 是 OASIS 组织安全服务技术委员会(Security Services Technical Committee)的产品。更多说明请参考 https://wiki.oasis-open.org/security/FrontPage。
- **MFA**：多因素认证（Multi factor authentication，包含 **Two-factor authentication** 双因子认证或 **2FA**），是一种计算机访问控制的方法，用户要通过两种及以上的认证机制验证通过之后，才能被授权使用计算机资源。MFA 的目的是建立一个多层次的防御，使未经授权的个体访问计算机系统或网络更加困难，从而提高系统安全性。
- **OTP**：一次性密码（英语：One Time Password，简称 OTP），又称动态密码或单次有效密码，是仅对计算机系统或其他数字设备上的一个登录会话或事务有效的密码。OTP 避免了与传统（静态）基于密码的身份验证相关的许多缺点。部分实现如：UKey、密码卡、手机验证码、邮箱验证码等。
- **TOTP**：Time-based One-time Password，基于时间的一次性密码
- **IAM**：身份和访问管理（Identity and access management，IAM 或 IdAM），也称为身份管理（Identity management，**IdM**），是一种策略和技术框架，用于确保企业中的成员能够适当地访问数据资源。近年来，随着合规要求变得越来越严格和复杂，身份和访问管理解决方案变得越来越普遍和重要。
- **IDaaS**：身份即服务，可以理解为云化版的 IAM。
- **LDAP**：轻量级目录访问协议（LDAP）是一种开放的、与供应商无关的，通过 Internet 协议（IP）网络访问和维护分布式目录信息服务的行业标准应用程序协议。
- **JustAuth**：一款开源的整合第三方登录的组件。目前已支持 Github、Gitee、微博、钉钉、百度、Coding、腾讯云开发者平台、OSChina、支付宝、QQ、微信、淘宝、Google、Facebook、抖音、领英、小米、微软、今日头条、Teambition、StackOverflow、Pinterest、人人、华为、企业微信、酷家乐、Gitlab、美团、饿了么和推特等第三方平台的授权登录。 Login, so easy!
- **Form-based：**基于表单的形式实现登录认证。

© 2020-2040 符节科技 版权所有

感谢阅读，谢谢支持！

符节科技 JustAuthPlus

© 2020-2040 符节科技 版权所有