

分布式身份 最佳实践



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

最佳实践

颁发毕业证

最佳实践

颁发毕业证

最近更新时间：2024-01-15 10:47:51

业务场景

近年来，职业培训的需求与日俱增，而职业培训通常以技能证书作为培训证明，因此，证书管理的规模也不断扩大。2021年底，人社部、教育部等部门联合发布了《“十四五”职业技能培训规划》，这是我国首次编制的国家级职业技能培训五年专项规划，代表我国对职业培训及其成果的重视。

除职业培训领域证书数量的增长外，高校毕业生人数（学历证书）也是屡创新高，据国家统计局数据显示，2022年全国高校毕业生将达到1076万，成为历史之最。如何完成毕业生证书的有效颁发和验证，成为了用人单位与求职毕业生之间的关键环节。

对个人而言，在应聘、考评等情况下，需对学历或毕业证书（学位证书）的真实性、合法性予以验证，避免假学历。对于企业，在项目投标、资质获取等情况下，也需对企业营业执照、能力证书、各种资质证书等进行鉴别，杜绝假资质。

因此，本文介绍如何快速实现一个基于 TDID 的颁发毕业证场景的区块链应用，以解决上述所提出的问题。本文以高校毕业证的颁发和验证为例，请结合您的业务场景参考相应的应用设计方法。

- [毕业证场景](#)
- [颁发毕业证](#)
- [验证毕业证](#)

应用设计

以下为毕业季时毕业证书颁发和验证所涉及的角色和业务流程。

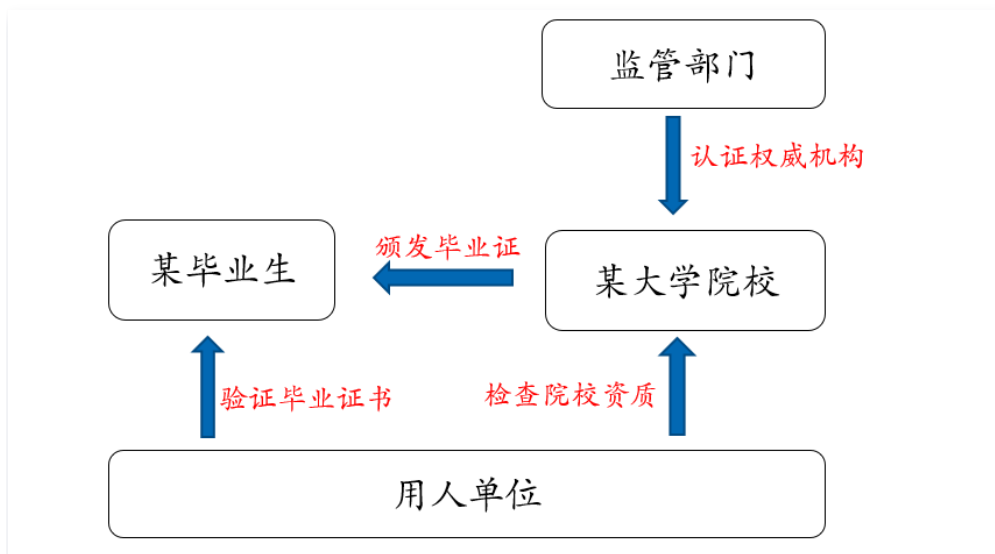
其中主要角色包括：

- 监管部门：例如 XX 省教育局
- 某大学院校：例如 XX 大学
- 用人单位：例如 XX 公司
- 某毕业生：例如 XX 大学 2022 年的某毕业生

主要业务流程包括：

- 部署应用：监管部门使用 TDID 控制台部署一个 DID 应用。
- 注册 DID：大学院校和毕业生，分别注册 DID 标识。
- 认证权威机构：监管部门将大学院校的 DID 标识认证为有教育资质的权威机构。
- 创建毕业证书模板：监管部门使用 TDID 控制台创建毕业证书模板。
- 签发凭证：权威机构为毕业生签发毕业证书凭证。
- 验证凭证：用人单位验证毕业证书凭证的有效性。

毕业证场景



业务流程：

1. 注册 DID：某大学院校（机构）和某毕业生注册 DID 标识。
2. 认证机构：监管者（DID 应用部署方）有权限把某大学院校认证为有教育资质的权威机构。
3. 签发凭证：某大学院校给某毕业生签发毕业凭证，凭证中包含毕业生 DID，凭证内容由其私钥签名。
4. 验证凭证：某毕业生出示毕业凭证，用人单位验证毕业生和机构 DID 以及数字签名。

颁发毕业证

1. 注册 DID

- 学生通过 DAPP 注册身份，尽可能减少暴露学生信息。
- 学校通过 DAPP 注册身份，尽可能公开学校信息。

涉及接口：[CreateTDidByHost](#)，[CreateTDidByPubKey](#)

2. 认证机构

- 监管方认证学校资质，用应用部署方的账号在 DID 平台把学校设置为权威机构。
- 监管方在 DID 平台注册毕业证的凭证模板，模板定义了毕业证的有效字段。

3. 颁发毕业证

- 学校获取学生的 DID 文档，验证学生身份。
- 学校获取毕业凭证模板，根据模板签发毕业证。

涉及接口：[GetTDidDocument](#)，[QueryCPT](#)，[IssueCredential](#)

验证毕业证

1. 学生出示毕业证

- 学生为了减少信息泄露如不暴露身份证号，可以用原始凭证生成选择性批露凭证。
- 使用选择性批露凭证出示给用户单位。

涉及接口：[CreateDisclosedCredential](#)

2. 用人单位验证毕业证

- 用人单位获取凭证声明的学生 DID 并解析，验证学生信息。
- 用人单位获取凭证颁发机构的 DID 并解析，验证机构信息。
- 用人单位获取凭证颁发机构 DID 的权威机构信息，验证机构资质。
- 用人单位验证毕业凭证是否符合模板和声明内容，验证凭证是否符合要求。
- 用人单位验证毕业凭证的签名，验证凭证是否由有相应资质的学校颁发。

涉及接口：[GetTDidDocument](#)，[QueryCPT](#)，[VerifyCredentials](#)

代码实践

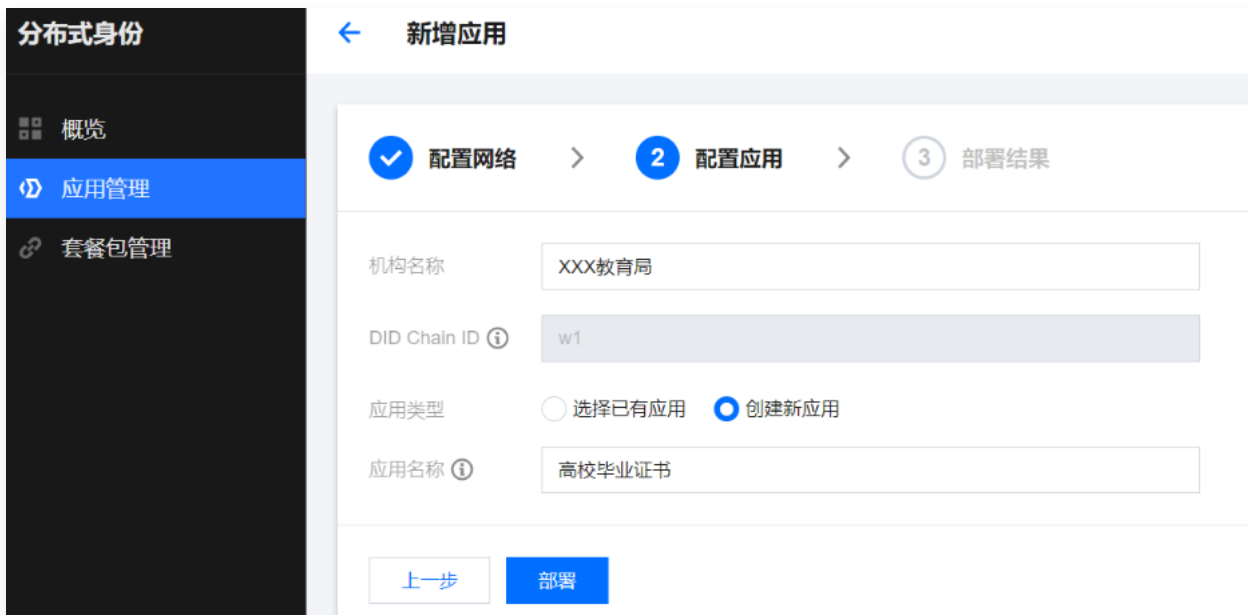
以下将结合代码，详细介绍如何完成上述应用场景的开发。

步骤 1：部署应用

1. 登录 TDID 控制台，在左侧导航中选择**应用管理**。
2. 单击**新增应用**。
3. 在配置网络中，选择 **WeCert 公共区块链网络**。如下图所示：



4. 在**配置应用**中，填写机构名称为“XX 省教育局”以及应用名称为“高校毕业证书”，并部署应用。如下图所示：



5. 单击部署。待部署成功后，将可以在应用列表看到该应用，及其应用 ID，系统将自动为“XX 省教育局”注册一个 DID 标识，并默认认证为权威机构。如下图所示：



步骤 2: 注册 DID

1. 单击 [tdid-diploma-demo.zip](#) 下载最新 demo 代码。
2. 从 [访问管理控制台](#) 获取用户的 SecretId 和 SecretKey、SDK 的访问域名以及上述的应用 ID，并配置到 demo 代码中。

```

15  const (
16      // TDID 配置项
17      // TDID控制台账户的SecretId和SecretKey
18      SecretId = "AKID71GMXXXXXXXXXXXXo1cEXXXXXXXXXXXXhQbj"
19      SecretKey = "ACXXXXXXXXXXXXXSSXXXXXXXXXXXXihjt2L"
20
21      // TDID访问域名
22      TDID_DOMAIN = "tdid.tencentcloudapi.com"
23
24      // TDID-高校毕业证书应用ID
25      DAPIID = uint64(233)
26
27      // 毕业证书凭证模板
28      CPTID = 1000
29  )
30
    
```

3. 为 XX 大学注册 DID，并将其学校名称作为自定义属性写到 DID 文档中，注册成功则可获得相应的 DID 标识。

```

func main() {
    universityDid, err := CreateUniversityDid("XX大学")
    if err != nil {
        panic(err)
    }
    fmt.Println("XX大学的DID标识为:", universityDid)
}

func CreateUniversityDid(universityName string) (string, error) {
    c := GetSdkClient()
    attr := make(map[string]string)
    attr["universityName"] = universityName
    customAttribute, _ := json.Marshal(attr)
    request := tdid.NewCreateTDidByHostRequest()
    request.SetScheme("http")
    request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
    // 设置应用ID
    request.DAPIID = common.Uint64Ptr(DAPIID)
    // 设置高校DID标识的自定义属性，此处包括学校名称
    request.CustomAttribute = common.StringPtr(string(customAttribute))
    response, err := c.CreateTDidByHost(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        return "", fmt.Errorf("An API error has returned: %v", err)
    }
    // 非SDK异常，直接失败。实际代码中可以加入其他的处理。
    if err != nil {
        return "", err
    }
    return *response.Response.Did, nil
}
    
```



```
• [root@VM-234-13-centos ~/code/tdid-diploma-demo]# go run main.go
XX大学的DID标识为: did:tdid:c117:0x0c1c75493fd114a3fce7567f192653ff45dd08cc
○ [root@VM-234-13-centos ~/code/tdid-diploma-demo]#
```

4. 为 XX 毕业生注册 DID，并将姓名和学号作为自定义属性写到 DID 文档中，注册成功则可获得相应的 DID 标识。

```
func main() {
    studentDid, err := CreateStudentDid("张三", "20220001")
    if err != nil {
        panic(err)
    }
    fmt.Println("XX毕业生的DID标识为:", studentDid)
}

func CreateStudentDid(studentName string, studentId string) (string, error) {
    c := GetSdkClient()
    attr := make(map[string]string)
    attr["studentName"] = studentName
    attr["studentId"] = studentId
    customAttribute, _ := json.Marshal(attr)
    request := tdid.NewCreateTDidByHostRequest()
    request.SetScheme("http")
    request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
    // 设置应用ID
    request.DAPIId = common.Uint64Ptr(DAPIID)
    // 设置毕业生DID标识的自定义属性，此处包括姓名和学号
    request.CustomAttribute = common.StringPtr(string(customAttribute))
    response, err := c.CreateTDidByHost(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        return "", fmt.Errorf("An API error has returned: %v", err)
    }
    // 非SDK异常，直接失败。实际代码中可以加入其他的处理。
    if err != nil {
        return "", err
    }
    return *response.Response.Did, nil
}
```

```
• [root@VM-234-13-centos ~/code/tdid-diploma-demo]# go run main.go
XX毕业生的DID标识为: did:tdid:c117:0xe456a62a5cf71bab70558d06971c3ee847a03941
○ [root@VM-234-13-centos ~/code/tdid-diploma-demo]#
```

步骤 3: 认证权威机构

在 TDID 控制台将 XX 大学的 DID 标识，认证为权威机构 XX 大学。

将 DID 认证为凭证发行的权威机构

机构DID: did:tdid:c117: [redacted]

机构名称: XX大学

备注 (选填): XX大学

[确定](#) [取消](#)

应用: 高校毕业证书

权威机构

[新建权威机构](#)

DIDs	权威机构	DID链上标签	认证时间	认证备注
did:tdid:c117:[redacted] did08cc	XX大学	0	2023-06-26 20:16:26	XX大学
did:tdid:c117:[redacted] ...ee1ab6	XX省教育厅	0	2023-06-26 17:05:14	XX省教育厅

左侧菜单: 基本信息, DID 管理 (DID 列表, 权威机构, DID 标签), 合约管理, 凭证模板管理, 披露策略管理, 应用概览

步骤 4: 创建毕业证书模板

毕业证书需要遵循一定的格式要求, 在本应用中, 将通过创建毕业证书模板, 来规定毕业证书的格式, 在后续验证凭证时, 将会校验毕业证书凭证是否满足模板的要求。

在 TDID 控制台, 新建毕业证书模板, 规定毕业证书必须包括: 高校名称(`universityName`)、毕业生姓名(`studentName`)、毕业生学号(`studentId`)、毕业生 DID 标识(`studentDid`)四个字段。创建成功, 则可获得该凭证模板的 Id, 即下图中的 **模板 (CPT) ID**, 本文示例为 1000。

新建模板 ×

✔ 选择类型 >
 2 定义模板 >
 3 注册上链

CPT名称 ✔

CPT描述 (选填) ✔

CPT ID (选填) ✔

数据结构定义 (凭证声明Claim的结构定义) [什么是Claim?](#)

1	<input type="text" value="studentId"/>	字符串	<input type="text" value="毕业生学号"/>	
2	<input type="text" value="studentName"/>	字符串	<input type="text" value="毕业生姓名"/>	
3	<input type="text" value="universityName"/>	字符串	<input type="text" value="高校名称"/>	
4	<input type="text" value="studentDid"/>	字符串	<input type="text" value="毕业生DID标识"/>	

Claim中的属性需要以字母开头

+ 新增字段

上一步
下一步

← 应用: 高校毕业证书

凭证模板管理

新建模板

模板 (CPT) ID	模板名称	模板描述	状态	模板类型
1000	毕业证书	毕业证书的格式要求	已启用	普通模板
1	operation	operate temp credential	已启用	系统模板

合约管理

凭证模板管理

披露策略管理

应用概览

步骤 5: 签发毕业凭证

在毕业证书模板创建之后，XX 大学则可为每一个毕业生颁发毕业证书凭证，具体如下所示。其中 XX 大学的 DID 标识作为颁发者，1000 作为使用的凭证模板 Id，凭证的声明内容包括高校名称(issuerName)、毕业生姓名(studentName)、毕业生学号(studentId)、毕业生 DID 标识(studentDid)四个字段，以确保格式符合毕业凭证模板要求。

```
func QueryAuthorityInfo(credentialData string) (string, error) {
    data := make(map[string]interface{})
    json.Unmarshal([]byte(credentialData), &data)
    issuerDid := data["issuer"].(string)
    c := GetSdkClient()
    request := tdid.NewQueryAuthorityInfoRequest()
    request.SetScheme("http")
    request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
    // 设置应用ID
    request.DAPId = common.Uint64Ptr(DAPID)
    request.Did = &issuerDid
    response, err := c.QueryAuthorityInfo(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        return "", fmt.Errorf("An API error has returned: %v", err)
    }
    // 非SDK异常，直接失败。实际代码中可以加入其他的处理。
    if err != nil {
        return "", err
    }
    return response.ToJsonString(), nil
}
```

```
func VerifyCredential(credentialData string) (string, error) {
    authorityInfo, err := QueryAuthorityInfo(credentialData)
    if err != nil {
        panic(err)
    }
    fmt.Println("颁发者权威机构信息为: ", authorityInfo)
    // 在验证之前，可取出credentialData中的issuer字段，判断调用
    c := GetSdkClient()
    request := tdid.NewVerifyCredentialsRequest()
    request.SetScheme("http")
    request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
    // 设置应用ID
    request.DAPId = common.Uint64Ptr(DAPID)
```

```
request.DAPIID = common.UInt64Ptr(DAPIID)
request.CredentialData = &credentialData
response, err := c.VerifyCredentials(request)
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    return "", fmt.Errorf("An API error has returned: %v", err)
}
// 非SDK异常, 直接失败。实际代码中可以加入其他的处理。
if err != nil {
    return "", err
}
return response.ToJsonString(), nil
}
```

毕业凭证的具体内容如下所示:

```
{
  "cptId": 1000,
  "issuer": "did:tdid:c117:0x0c1c75493fd114a3fce7567f192653ff45dd08cc",
  "expirationDate": "2050-12-01T10:00:00+08:00",
  "issuanceDate": "2023-06-27T15:22:16+08:00",
  "context": "https://github.com/TencentCloud-Blockchain/TDID/blob/main/context/v1",
  "id": "266ab6d51f0e27fdb702c620f79d7d43",
  "type": ["VerifiableCredential"],
  "credentialSubject": {
    "studentDid": "did:tdid:c117:0xe456a62a5cf71bab70558d06971c3ee847a03941",
    "studentId": "20220001",
    "studentName": "张三",
    "universityName": "XX大学"
  },
  "proof": {
    "created": "2023-06-27T15:22:16+08:00",
    "creator": "did:tdid:c117:0x0c1c75493fd114a3fce7567f192653ff45dd08cc#keys-0",
    "signatureValue":
"MEQCIEh4uNJMWyHgSlnRaTjNvgiYIFd01xxOCsiMCimXtP0MAiAVwd/R2CJ0imrJRuhvI0Tf
qwb60SJ7HOjG3EwFmPonLA==",
    "type": "Secp256r1",
    "metaDigest":
"12cff20ba5132a1d8e7ddfc5173959b2fa50b3ca70ce2cd2b7cfd81172e73cd9",
    "vcDigest":
"a7162f47a0daa58ac79ec3bc706c17626669787853875e5458ad5ef7f37dd6dd",
    "privacy": "Public",
    "salt": {
      "studentDid": "I9o7Y",
      "studentId": "H2KVI",

```

```
"studentName": "24zSu",
"universityName": "D8eqW"
}
}
}
```

步骤 6: 验证毕业凭证

在 XX 毕业生在拿到由 XX 大学颁发的毕业证书凭证之后，则可以交给用人单位，由用人单位使用以下代码进行验证。验证环节首先从凭证内容中取出颁发者 DID 标识，查询该 DID 标识是否已经注册为权威机构，之后再调用 VerifyCredentials 接口，验证凭证的有效性（是否由 XX 大学签发）。

```
func QueryAuthorityInfo(credentialData string) (string, error) {
    data := make(map[string]interface{})
    json.Unmarshal([]byte(credentialData), &data)
    issuerDid := data["issuer"].(string)
    c := GetSdkClient()
    request := tdid.NewQueryAuthorityInfoRequest()
    request.SetScheme("http")
    request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
    // 设置应用ID
    request.DAPId = common.Uint64Ptr(DAPID)
    request.Did = &issuerDid
    response, err := c.QueryAuthorityInfo(request)
    if _, ok := err.(*errors.TencentCloudSDKError); ok {
        return "", fmt.Errorf("An API error has returned: %v", err)
    }
    // 非SDK异常，直接失败。实际代码中可以加入其他的处理。
    if err != nil {
        return "", err
    }
    return response.ToJsonString(), nil
}

func VerifyCredential(credentialData string) (string, error) {
    authorityInfo, err := QueryAuthorityInfo(credentialData)
    if err != nil {
        panic(err)
    }
    fmt.Println("颁发者权威机构信息为: ", authorityInfo)
    // 在验证之前，可取出credentialData中的issuer字段，判断调用
    c := GetSdkClient()
}
```

```
request := tdid.NewVerifyCredentialsRequest()
request.SetScheme("http")
request.WithApiInfo(strings.Split(TDID_DOMAIN, ".")[0], request.GetVersion(), request.GetAction())
// 设置应用ID
request.DAPIId = common.Uint64Ptr(DAPIID)
request.CredentialData = &credentialData
response, err := c.VerifyCredentials(request)
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    return "", fmt.Errorf("An API error has returned: %v", err)
}
// 非SDK异常, 直接失败。实际代码中可以加入其他的处理。
if err != nil {
    return "", err
}
return response.ToJsonString(), nil
}
```

验证结果如下所示：其中 VerifyCode 为 0，且 VerifyMessage 为 success，则说明验证成功。

```
颁发者权威机构信息为: {"Response":{"Name":"XX大学","Did":"did:tdid:c117:0x0c1c75493fd114a3fce7567f192653ff45dd08cc","Status":1,"Description":"XX大学","RecognizeTime":"2023-06-26 20:16:26","RequestId":"867d2f5d-bcae-439d-8084-ca20ca02ad34"}}
毕业证书凭证验证结果为 {"Response":{"Result":true,"VerifyCode":0,"VerifyMessage":"success","RequestId":"12c61717-6eab-4da0-94c8-7b8a360941c6"}}
```