Cloudbase Tutorials

Web starter

-- anonymous-login/文件夹

匿名登陆，和Firebase的匿名登陆类似

-- auth/文件夹

1. 创建云函数并且开通HTTP访问
   `cloudbase service:create -p /generateTicket -f generateTicket`
2. 部署静态页面
   `cloudbase hosting:deploy ./static ./auth`

   第一个相对路径是本地路径，第二个是服务器端路径。部署完成后可以通过`https://你的子域名.tcloudbaseapp.com/auth`来访问。

   会Firebase CLI的可以参考Firebase的静态页面托管
3. 几个常见的API
   1. 初始化

```
// 初始化 CloudBase
this.app = tcb.init({
  env: envId
})
this.auth = this.app.auth({
  persistence: "local"
})
```

   2. 匿名登陆

```
this.auth
  .anonymousAuthProvider()
  .signIn()
  .then(() => {
    window.alert("[匿名登录] 登录成功")
  })
  .catch((err) => {
    if (err.message === "[INVALID_OPERATION] Invalid operation") {
      window.alert("不能从 [自定义登录] 或 [微信登录] 退化到 [匿名登录]")
    } else {
      throw err
```

```
    }
  })
```

3. 自定义登陆

```
var userId = this.customUserId.value
// 校验自定义用户ID
var checkUserIdRegex = /^[a-zA-Z0-9_\-#@~=*(){}[\]:.,<>+]{4,32}$/
if (!checkUserIdRegex.test(userId)) {
  window.alert("[自定义登录] 自定义用户ID必须是4-32位字符, 且字符只能是大小写英文字母、数字、以及 _-#@(){}<>[]:.,<>+#~ 中的字符")
  return
}
// 换取自定义登录凭证
axios.post(
  "https://" + envId + ".service.tcloudbase.com/generateTicket",
  { customUserId: userId}
)
.then((res) => {
  // 进行自定义登录
  this.auth
    .customAuthProvider()
    .signIn(res.data.ticket)
    .then(() => {
      window.alert("[自定义登录] 登录成功")
    })
    .catch((err) => {
      throw err
    })
})
.catch((err) => {
  throw err
})
```

4. 退出登录

```
// 退出登录
signOut(e) {
  e.stopPropagation()
  e.preventDefault()

  this.auth
    .signOut()
    .then(() => {
      window.alert("退出登录成功")
    })
    .catch((err) => {
      if (err.message === "[tcb-js-sdk] 匿名用户不支持登出操作") {
        window.alert("匿名用户不支持登出操作")
```

```
      } else {
        throw err
      }
    })
}
```

5. 更新数据

```
var loginState = this.auth.hasLoginState()
var nickName = this.customUserName.value
loginState
  .user
  .update({
    nickName: nickName,
  })
  .then(() => {
    window.alert("修改用户信息成功")
    this.onLoginStateChanged(loginState)
  })
  .catch((err) => {
    throw err
  })
```

--functions/文件夹

一个执行简单加法的云函数

- 注意：`cloudbase function:deploy`可以选择需要部署的函数名称，默认需要在`cloudbaserc.js`里声明，比如

```
functions: [
  {
      name: "addNumbers",
      // 超时时间
      timeout: 5,
      // 环境变量
      envVariables: {},
      runtime: "Nodejs10.15",
      // 内存 128
      memorySize: 128,
      handler: "index.main"
  }
]
```

- 函数声明格式

```
exports.main = async (event) => {
  console.log(event);
  return {
    firstNumber: event.firstNumber,
    secondNumber: event.secondNumber,
    operationResult: event.firstNumber + event.secondNumber
  };
}
```

- 函数调用方式

```
this.app.callFunction({
  name: "addNumbers",
  data: { firstNumber: firstNumber, secondNumber: secondNumber }
})
```

- 示例：https://qcloud-0gcp0fz3bfc69053-1256664429.tcloudbaseapp.com/functions

--database/文件夹

- 增删(改)查

```
var coll = this.app.database().collection("test_db")
// 增，只返回状态码
coll.add({
  name: name,
  score: score
}).then({ code })

// 删，只返回状态码
coll.where({
  name: name
}).remove().then((res)

// 查，返回状态码和data字段
coll.where({
  name: name
}).get().then(({ code, data}) => {})

coll.where({}).get() // 查询全部

coll.where({}).orderBy("score", "desc") // 排序
```

--storage/文件夹

建一个文件input，以下代码会上传到云存储，格式同Firebase Storage

```
var file = e.target.files[0]
var name = file.name

this.app.uploadFile({
  cloudPath: name,
  filePath: file
}).then(res => {
  // 云文件ID
  var fileID = res.fileID
  // 通过云文件ID 获取 云文件链接
  this.app.getTempFileURL({
    fileList: [fileID]
  }).then(res2 => {
    var fileObj = res2.fileList[0]
    var url = fileObj.tempFileURL
    document.getElementById('linkbox').innerHTML = `<a href=${url}>
${name} </a>`
  })
})
```

Misc

如何使用sed替换

- MacOS
- Unix