TESLA

特斯拉超充减免第三方开发规范文档

Table of Contents

版本描述

Version	Modified By	Date	Description
1.0.0	施彭浩	2021.12.06	第一版文档
1.0.1	施彭浩	2021.12.09	修改 postman 脚本不合法的问题，添加鉴权步骤，基于 service account 的判断

1. 文档描述

1. 本文档提供给与特斯拉对接的第三方停车系统，第三方需要按照文档规定的接口规范和接口认证鉴权方式进行开发
2. 接口开发完成后，特斯拉会调用第三方开发好的接口将超充减免信息发送给第三方。
3. 若不做特殊说明，接口调用统一采用 http post 的调用方式
4. 接口安全认证统一采用 oauth2.0 的鉴权方式，特斯拉调用第三方接口时，会在请求 header 中携带 token，第三方只需要验证该 token 的合法性，若 token 合法处理请求，特斯拉会提供鉴权示例代码。

2. 接口规范

2.1. 第三方开放 url

- url: http://<第三方 uri>/api/suc-coupon

2.2. 请求头

Content-Type: application/json

Authorization:Bearer <特斯拉携带的 token>

2.3 请求字段

参数	类型	是否必须	名称	备注
orderId	String	Y	订单 ID	特斯拉生成的全局唯一减免订单号
parkId	String	N	停车场 ID	第三方停车场 ID，如需填写，第三方需提前提供相关数据
parkName	String	N	停车场名称	第三方停车场名称，如需填写，第三方需提前提供相关数据
plateNumber	String	Y	车牌号	充电车的车牌号

discountType	String	Y	减免类型	当 discontType 为 0 时，代表时间减免，为 1 时，代表金额减免,为 2 时，代表时间和金额都减免
duration	String	N	减免时间	discountType 类型为 0 或 2 时，为必填项，单位是分钟
amount	String	N	减免金额	discountType 类型为 1 或 2 时，为必填项，单位是元，到小数点后两位
startTime	String	Y	充电起始时间	插入充电桩的时间
endTime	String	Y	充电中止时间	拔出充电桩的时间

示例:


```
{
        "orderId": " TESLAORDER19961225",
        "parkId": "1009033",
        "parkName": "XX 停车场",
        "plateNumber": "沪 A111111",
        "discountType": "2",
        "duration": " 60",
        "startTime": "2021-11-23 10:30:00",
        "endTime": "2021-11-23 11:30:00"
}
```

2.4 请求回复字段

参数	类型	是否必须	名称	备注
statusCode	String	Y	业务状态码	当状态码为 0 时，代表请求成功，当状态码为 1 时，代表减免订单处理失败
Msg	String	Y	处理信息	请求处理信息，处理失败时，返回失败信息。

示例 1:


```
{
        "statusCode": "0",
        "msg": "请求成功"
}
```


示例 2:


```
{
        "statusCode": "1",
        "msg": "车牌号不存在"
}
```

2.5 Postman 完整示例

```
curl --location --request POST 'http://<第三方 uri>/api/suc-coupon' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJYcTZybi1YWkVaRkl0QXh2NUZxczVXbHF4UVp5VnVheTJBVEZLcUZubmdnIn0.eyJle
HAiOjE2Mzg4NDA1MzIsImlhdCI6MTYzNzYzMDkzMiwiYXV0aF90aW1lIjoxNjM3NjMwOTMyLCJqdGkiOiI1YjRlZGM3NS01MDM4LTRiZDEtYTQwNi1mZj
NmNjlmZWVlNTEiLCJpc3MiOiJodHRwczovL2JqbS1zdGcudGVzbGEuY24vYXV0aC9yZWFsbXMvV29ya2Zsb3ciLCJhdWQiOiJhY2NvdW50Iiwic3ViIjo
iMWI5NGFkMzktMzk3Ni00MWJiLWIxNDgtNGI0OGFjMmNjY2JhIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiZmxvd2FibGUtY2xpZW50Iiwic2Vzc2lvbl9z
dGF0ZSI6IjU2NjZlYTZiLTg3YzMtNGIzZS04YjFlLWY1NjljMDExNzQ5YiIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsiQkpNLkRlc
2suVXNlciIsIkJKTS5TRFMuVXNlciIsIkJKTS5TRFMuTG9jYXRpb24uQWRtaW4iLCJCSk0uR2lnYUZhY2lsaXR5LlVzZXIiLCJCSk0uUm9hZE1hbmFnZW
1lbnQuVXNlciIsInZpZXctYWxsLXRhc2siLCJCSk0uU0RTLkNvbnRyYWN0LkFkbWluIiwiQkpNLlN1Q09uYm9hcmQuVXNlciIsIkJKTS5EQ1IuVXNlciI
sIkJKTUludGVybmFsLlN5c3RlbUFkbWluLk1vYmlsZSIsIkJKTS5NYXAuVXNlciIsIm9mZmxpbmVfYWNjZXNzIiwiYWNjZXNzLXRhc2siLCJhY2Nlc3Mt
bW9kZWxlciIsInVtYV9hdXRob3JpemF0aW9uIiwiQkpNLlNEUy5Mb2NhdGlvbi5WaWV3QWxsIl19LCJyZXNvdXJjZV9hY2Nlc3MiOnsiYWNjb3VudCI6e
yJyb2xlcyI6WyJtYW5hZ2UtYWNjb3VudCIsIm1hbmFnZS1hY2NvdW50LWxpbmtzIiwidmlldy1wcm9maWxlIl19fSwic2NvcGUiOiJvcGVuaWQgbWljcm
9wcm9maWxlLWp3dCBwcm9maWxlIHBob25lIG9mZmxpbmVfYWNjZXNzIGFkZHJlc3MgZW1haWwiLCJ1c2VyR3JvdXBzIjpbIkJKTS5EQ1IuVFBNIiwiQkp
NLkRlcGFydG1lbnQuVFBNIiwiQkpNLlNEUy5Db250cmFjdFJldmlld2VyLlRheCIsIkJKTS5UZXNsYS5Vc2VyIiwiQkpNLldvcmtzaG9wLlRQTSIsIkVI
U1Blcm1pdEFyZWFQSUMiLCJFSFNQZXJtaXRQSUMiXSwidXBuIjoiMWE5ZjUxOWMtMTM5Ny00MjU4LWIyMWUtODIxZWE0NDAzMzBjIiwiZW1haWxfdmVya
WZpZWQiOmZhbHNlLCJhZGRyZXNzIjp7fSwibmFtZSI6IlBlbmdoYW8gU2hpICjmlr0g5b2t5rWpKSIsImdyb3VwcyI6WyJCSk0uRGVzay5Vc2VyIiwiQk
pNLlNEUy5Vc2VyIiwiQkpNLlNEUy5Mb2NhdGlvbi5BZG1pbiIsIkJKTS5HaWdhRmFjaWxpdHkuVXNlciIsIkJKTS5Sb2FkTWFuYWdlbWVudC5Vc2VyIiw
idmlldy1hbGwtdGFzayIsIkJKTS5TRFMuQ29udHJhY3QuQWRtaW4iLCJCSk0uU3VDT25ib2FyZC5Vc2VyIiwiQkpNLkRDUi5Vc2VyIiwiQkpNSW50ZXJu
YWwuU3lzdGVtQWRtaW4uTW9iaWxlIiwiQkpNLk1hcC5Vc2VyIiwib2ZmbGluZV9hY2Nlc3MiLCJhY2Nlc3MtdGFzayIsImFjY2Vzcy1tb2RlbGVyIiwid
W1hX2F1dGhvcml6YXRpb24iLCJCSk0uU0RTLkxvY2F0aW9uLlZpZXdBbGwiXSwicHJlZmVycmVkX3VzZXJuYW1lIjoiMWE5ZjUxOWMtMTM5Ny00MjU4LW
IyMWUtODIxZWE0NDAzMzBjIiwiZ2l2ZW5fbmFtZSI6IlBlbmdoYW8iLCJmYW1pbHlfbmFtZSI6IlNoaSIsImVtYWlsIjoicGVuZ2hzaGlAdGVzbGEuY29
tIn0.LXU6BwOM29c8GG1nQDirThYg28W_Q9J_huqceXQSmiZsd11eg7d3X412jv1yKkm2NQMpw_aQGWW30T0MJhpR-
c7ss3uTJomUaMzs3YRbYYY5bYqcKWKQwHqUbEdWYki7beE4PiGeTq_dclovEPMcKc8VBrso2LkrqgfaQaDDivYkxAQJSNETEMrfwb0S1uqPeLEy0t-
AVpIjyJppibL-CbGedxXk3leV4YPB_EXIQ73cXiQmC8BhPYh5TWCsRUQTo1eaEmmbQmZS-
_VSmGCkaR06EjGidqKgSDbDr7xWtrhJ6CbKQuX9vpdJtDPbBkri6F5KRoS-Hew2AnayaC1M5w' \

--header 'Content-Type: application/json' \

--data-raw '{

    "orderId": " TESLAORDER19961225",

    "parkId": "1009033",

    "parkName": "XX 停车场",

    "plateNumber": "沪 A111111",

    "discountType": "2",

    "duration": " 60",

    "startTime": "2021-11-23 10:30:00",

    "endTime": "2021-11-23 11:30:00"

}

'
```

3. 鉴权方式

3.1. 鉴权描述

特斯拉会提供给第三方用于鉴权的 public key,当发送给第三方请求时，第三方只需要利用公钥对 token 的合法性进行检验，当 token 合法时，再去判断 token 中 payload 的 "preferred_username"字段是否为特斯拉提供给第三方的 service account，若鉴权通过，再去处理该请求，针对于每个不同的第三方，特斯拉会创建不同的 service account 供第三方使用。

- 示例 Public Key

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmeCR1cbXnh5JaSSsKNUQ
KuJdtykFm0xIXdZ8RoPFvKT9InH5SC+DzeruXzxQBaajpN5PEtTGounbJFIpD0PE
QAsCB7KAqe6zmjGyFuAyLiJxOtGGbTBBLkTI0jHXb3BaN9ojHc1LYjRzojDmhzCX
gAdTgIH/G5+brEQcFulBZ9KmzR8UI+/WDv7TnAlcR5umA77eJrYCgaVL/no7uhzf
lJDIwmG6BRukPLjYDZGT0yQcr/pNoeSQB+Of2INnE0H42u1YRvL+SRpF/3mo81zT
LzdHcjHWLUxqns9NBqODnIv2TDgd6qotMPKe4K0AxBLQNRDR71KErEL74WT7u33N
+wIDAQAB
-----END PUBLIC KEY-----
```

- 示例 Token

```
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJLeUlfS2NaakJWWEZtS1BmcXdBb
3Z5bUloTHNVWTVyTEhoNFFhM05McXNBIn0.eyJleHAiOjE2MzkwNjU1NzQsImlhdCI6MTYzOTAzNj
c3NCwianRpIjoiZjM0MmMxOGYtMmRmYy00ODE3LTk1YmYtODYwNWMzMzc5NTE3IiwiaXNzIjoiaHR
0cHM6Ly9iam0udGVzbGEuY24vYXV0aC9yZWFsbXMvV29ya2Zsb3ciLCJhdWQiOiJhY2NvdW50Iiwi
c3ViIjoiNzIzY2VmNmYtYmE0Ni00N2I0LWFjZWEtMGQ3NDM1MWM5Y2QyIiwidHlwIjoiQmVhcmVyI
iwiYXpwIjoidmVuZG9yLXhtIiwiYWNyIjoiMSIsInJlYWxtX2FjY2VzcyI6eyJyb2xlcyI6WyJvZm
ZsaW5lX2FjY2VzcyIsInVtYV9hdXRob3JpemF0aW9uIl19LCJyZXNvdXJjZV9hY2Nlc3MiOnsiYWN
jb3VudCI6eyJyb2xlcyI6WyJtYW5hZ2UtYWNjb3VudCIsIm1hbmFnZS1hY2NvdW50LWxpbmtzIiwi
dmlldy1wcm9maWxlIl19LCJ2ZW5kb3IteG0iOnsicm9sZXMiOlsidW1hX3Byb3RlY3Rpb24iXX19L
CJzY29wZSI6InByb2ZpbGUgZW1haWwiLCJjbGllbnRIb3N0IjoiMzIuNjAuNTcuMjMwIiwiZW1haW
xfdmVyaWZpZWQiOmZhbHNlLCJjbGllbnRJZCI6InZlbmRvci14bSIsInByZWZlcnJlZF91c2VybmF
tZSI6InNlcnZpY2UtYWNjb3VudC12ZW5kb3IteG0iLCJjbGllbnRBZGRyZXNzIjoiMzIuNjAuNTcu
MjMwIn0.e2bRo1xnU-MFqZlu6YZR_XGz2eWR9-
_YSLAHtOPCkLsf0inYfWRH_cFzwaHLQNm4kl3Q3KSuME5p3YxgvKDNSzOpvvgfpzjd74H7TXdKvzH
kYqjlMpKJgSJV7Z1xI3p-
iR3aSFlCvoYQeuFy8jQjhjp5RJPZh6pKhBT2iF6Xhxv4406bB92InmHj4YBawkR-
SrjLnnwPLzSrWtFFvPe9L9RoDHcp2JZccOV41KgBh3BLISvY2UTMVH7RwQsVva-
30TeRY279jxyD4tIGmDVEjiM1lHub6sdjME80ezsC-
jv_7inSzlWW5_qq16W3Ndkol_7TJHuM6N_Q4WqUgyNE5A
```

- 示例 Service Account

```
service-account-vendor-xm
```

3.2 鉴权示例

Golang 版本：

```
依赖于第三方库："github.com/dgrijalva/jwt-go/v4"
```

代码：

```
package main

import (
    "crypto/rsa"
    "crypto/x509"
    "encoding/pem"
    "errors"
    "fmt"
    "github.com/dgrijalva/jwt-go/v4"
    "strings"
)


var PublicKey1 string = `-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmeCR1cbXnh5JaSSsKNUQ
KuJdtykFm0xIXdZ8RoPFvKT9InH5SC+DzeruXzxQBaajpN5PEtTGounbJFIpD0PE
QAsCB7KAqe6zmjGyFuAyLiJxOtGGbTBBLkTI0jHXb3BaN9ojHc1LYjRzojDmhzCX
gAdTgIH/G5+brEQcFulBZ9KmzR8UI+/WDv7TnAlcR5umA77eJrYCgaVL/no7uhzf
lJDIwmG6BRukPLjYDZGT0yQcr/pNoeSQB+Of2INnE0H42u1YRvL+SRpF/3mo81zT
LzdHcjHWLUxqns9NBqODnIv2TDgd6qotMPKe4K0AxBLQNRDR71KErEL74WT7u33N
+wIDAQAB
-----END PUBLIC KEY-----`
var ExampleToken1 string =
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJLeUlfS2NaakJWWEZtS1BmcXdB
b3Z5bUloTHNVWTVyTEhoNFFhM05McXNBIn0.eyJleHAiOjE2MzkwNjU1NzQsImlhdCI6MTYzOTAzN
jc3NCwianRpIjoiZjM0MmMxOGYtMmRmYy00ODE3LTk1YmYtODYwNWMzMzc5NTE3IiwiaXNzIjoiaH
R0cHM6Ly9iam0udGVzbGEuY24vYXV0aC9yZWFsbXMvV29ya2Zsb3ciLCJhdWQiOiJhY2NvdW50Iiw
ic3ViIjoiNzIzY2VmNmYtYmE0Ni00N2I0LWFjZWEtMGQ3NDM1MWM5Y2QyIiwidHlwIjoiQmVhcmVy
IiwiYXpwIjoidmVuZG9yLXhtIiwiYWNyIjoiMSIsInJlYWxtX2FjY2VzcyI6eyJyb2xlcyI6WyJvZ
mZsaW5lX2FjY2VzcyIsInVtYV9hdXRob3JpemF0aW9uIl19LCJyZXNvdXJjZV9hY2Nlc3MiOnsiYW
Njb3VudCI6eyJyb2xlcyI6WyJtYW5hZ2UtYWNjb3VudCIsIm1hbmFnZS1hY2NvdW50LWxpbmtzIiw
idmlldy1wcm9maWxlIl19LCJ2ZW5kb3IteG0iOnsicm9sZXMiOlsidW1hX3Byb3RlY3Rpb24iXX19
LCJzY29wZSI6InByb2ZpbGUgZW1haWwiLCJjbGllbnRIb3N0IjoiMzIuNjAuNTcuMjMwIiwiZW1ha
WxfdmVyaWZpZWQiOmZhbHNlLCJjbGllbnRJZCI6InZlbmRvci14bSIsInByZWZlcnJlZF91c2Vybm
FtZSI6InNlcnZpY2UtYWNjb3VudC12ZW5kb3IteG0iLCJjbGllbnRBZGRyZXNzIjoiMzIuNjAuNTc
uMjMwIn0.e2bRo1xnU-MFqZlu6YZR_XGz2eWR9-
_YSLAHtOPCkLsf0inYfWRH_cFzwaHLQNm4kl3Q3KSuME5p3YxgvKDNSzOpvvgfpzjd74H7TXdKvzH
kYqjlMpKJgSJV7Z1xI3p-
iR3aSFlCvoYQeuFy8jQjhjp5RJPZh6pKhBT2iF6Xhxv4406bB92InmHj4YBawkR-
SrjLnnwPLzSrWtFFvPe9L9RoDHcp2JZccOV41KgBh3BLISvY2UTMVH7RwQsVva-
30TeRY279jxyD4tIGmDVEjiM1lHub6sdjME80ezsC-
jv_7inSzlWW5_qq16W3Ndkol_7TJHuM6N_Q4WqUgyNE5A"
```

```go
func VerifyToken(accessToken string, publicKeyStr *rsa.PublicKey,
serviceAccount string) error {

   token, err := jwt.Parse(accessToken, func(token *jwt.Token) (interface{},
error) {
      return publicKeyStr, nil
   }, jwt.WithoutAudienceValidation())
   if err != nil {
      switch err.(type) {
      case *jwt.TokenExpiredError:
         return errors.New("token expired")
      case *jwt.TokenNotValidYetError:
         return errors.New("token not valid yet")
      case *jwt.InvalidIssuerError:
         return errors.New("token issuer is invalid")
      default:
         return err
      }
   }
   if claims, ok := token.Claims.(jwt.MapClaims); ok && token.Valid {
      fmt.Println(claims["preferred_username"])
      if claims["preferred_username"] == serviceAccount {
         return nil
      } else {
         return errors.New("service account is not right")
      }
   }
   return errors.New("invalid token")
}


func parseRSAPublicKey(publicKey string) (*rsa.PublicKey, error) {
   data, _ := pem.Decode([]byte(strings.Trim(publicKey, " ")))
   publicKeyImported, err := x509.ParsePKIXPublicKey(data.Bytes)
   if err != nil {
      return nil, err
   }
   if rsaPub, ok := publicKeyImported.(*rsa.PublicKey); ok {
      return rsaPub, nil
   }
   return nil, fmt.Errorf("%+v is not a *rsa.PublicKey", publicKeyImported)
}


func main() {
   rsaPublicKey, _ := parseRSAPublicKey(PublicKey1)
   err := VerifyToken(ExampleToken1, rsaPublicKey, "service-account-vendor-
xm")
   if err != nil {
      fmt.Println("鉴权失败", err)
   } else {
      fmt.Println("鉴权成功")
   }
}
```

Java 版本：

仓库地址：

https://gitee.com/LostInAurora/java-token-verify