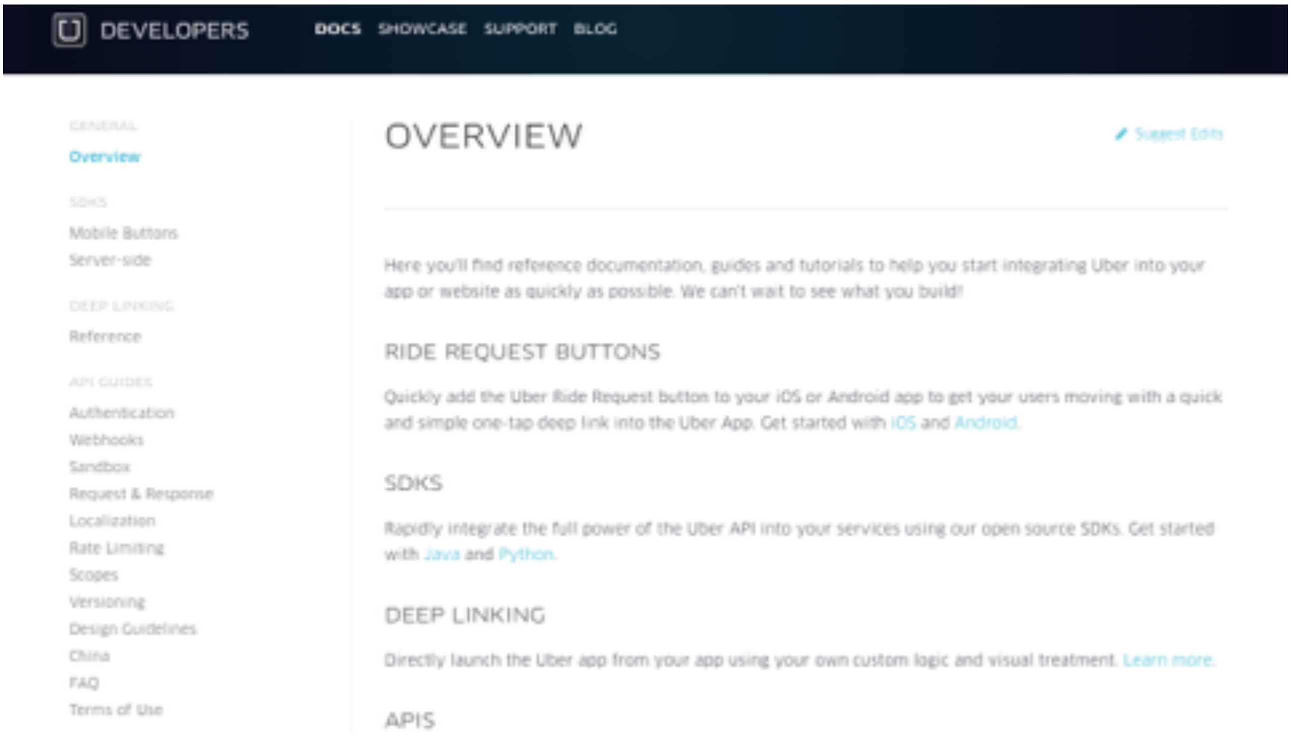


OVERVIEW

Here you'll find reference documentation, guides and tutorials to help you start integrating Uber into your app or website as quickly as possible. We can't wait to see what you build!

RIDE REQUEST BUTTONS

Quickly add the Uber Ride Request button to your iOS or Android app to get your users moving with a quick and simple one-tap deep link into the Uber App. Get started with iOS and Android.

SDKS

Rapidly integrate the full power of the Uber API into your services using our open source SDKs. Get started with Java and Python.

DEEP LINKING

Directly launch the Uber app from your app using your own custom logic and visual treatment. Learn more.

APIS

OVERVIEW
概览

Here you'll find reference documentation, guides and tutorials to help you start integrating Uber into your app or website as quickly as possible. We can't wait to see what you build!
在这里，你能迅速找到帮助你将Uber服务集成到你的app或网站当中所需要的参考文档、设计规范、开发指导。我们十分期待看到你的成果！

RIDE REQUEST BUTTONS
请求行程按钮

Quickly add the Uber Ride Request button to your iOS or Android app to get your users moving with a quick and simple one-tap deep link into the Uber App. Get started with iOS and Android.
快速添加Uber请求行程按钮到你的iOS或者Android应用程序当中，让你的用户可以通过一次点击快速链接到Uber应用程序当中。从iOS和Android两个部分开始吧。

SDKS

SDK

Rapidly integrate the full power of the Uber API into your services using our open source SDKs. Get started with Java and Python.

用我们的开源SDK，可以快速将Uber API的全部强大功能集成到你的服务当中。从Java和Python开始吧。

DEEP LINKING
深度链接

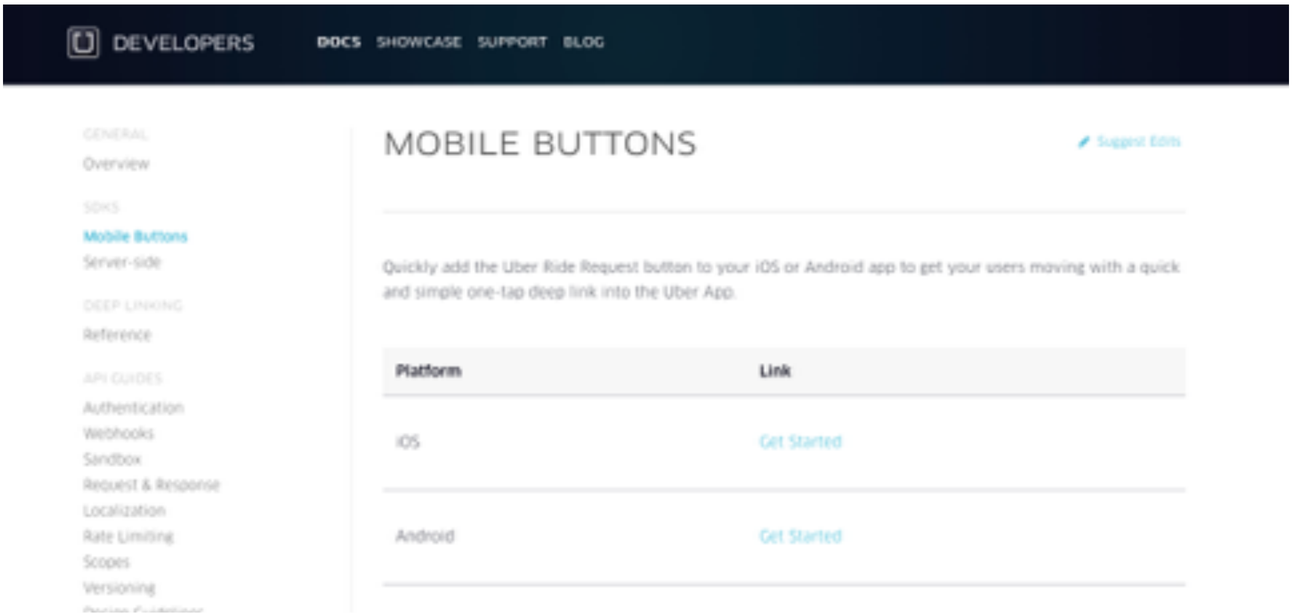
Directly launch the Uber app from your app using your own custom logic and visual treatment. Learn more.

可以直接从你的应用当中，按照你的用户逻辑和视觉体验，启动Uber应用。查看更多相关内容（Learn more）。

APIS
API

Need to go even deeper? The Rides API gives you direct access to the REST APIs that power our SDKs and Buttons. Learn more.

想要更深入整合？用车行程API可以给你直接访问为我们的SDK和“请求行程按钮”提供服务的REST API的能力。查看更多相关内容Learn more。



MOBILE BUTTONS

Quickly add the Uber Ride Request button to your iOS or Android app to get your users moving with a quick and simple one-tap deep link into the Uber App.

Platform	Link
iOS	Get Started
Android	Get Started

MOBILE BUTTONS

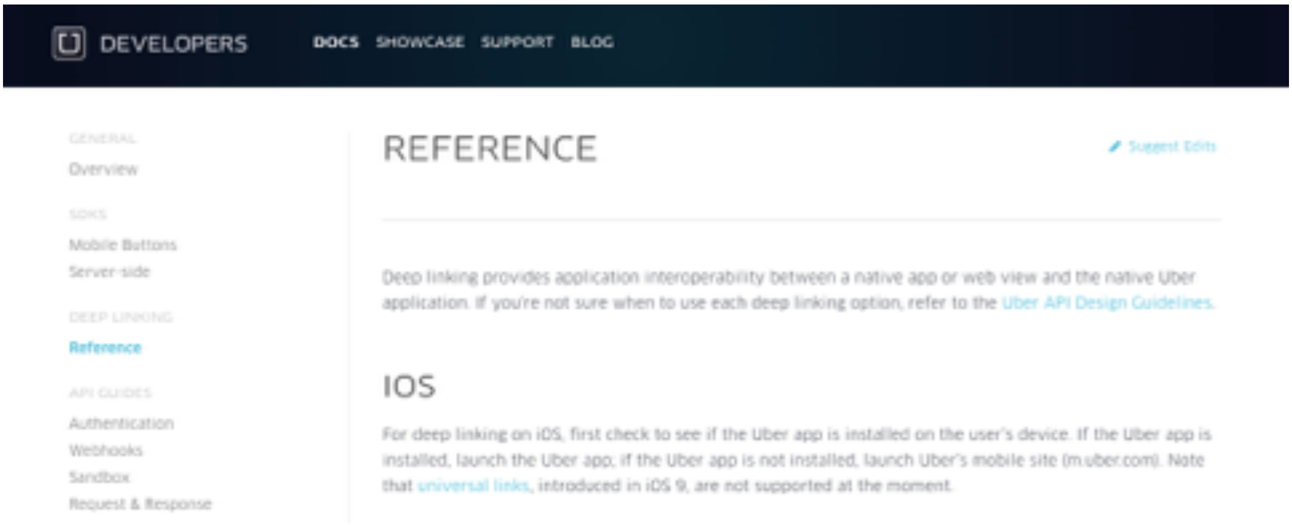
移动按钮

Quickly add the Uber Ride Request button to your iOS or Android app to get your users moving with a quick and simple one-tap deep link into the Uber App.
快速添加Uber请求行程按钮到你的iOS或者Android应用程序当中，让你的用户可以通过一次点击快速链接到Uber应用程序当中。

平台	链接
iOS	由此开始：https://github.com/uber/rides-ios-sdk/blob/master/README.md
Android	由此开始：https://github.com/uber/rides-android-sdk/blob/master/README.md

SERVER-SIDE
服务侧

语言	链接
Java	由此开始：https://github.com/uber/rides-java-sdk/blob/master/README.md
Python	由此开始：https://github.com/uber/rides-python-sdk/blob/master/README.rst



REFERENCE

Deep linking provides application interoperability between a native app or web view and the native Uber application. If you're not sure when to use each deep linking option, refer to the Uber API Design Guidelines.

IOS

For deep linking on iOS, first check to see if the Uber app is installed on the user's device. If the Uber app is installed, launch the Uber app; if the Uber app is not installed, launch Uber's mobile site (m.uber.com). Note that universal links, introduced in iOS 9, are not supported at the moment.

REFERENCE
参考文档

Deep linking provides application interoperability between a native app or web view and the native Uber application. If you're not sure when to use each deep linking option, refer to the Uber API Design Guidelines.
深度链接为应用程序提供了在原生app、web view框架和原生Uber客户端之间间进行互相操作的可能性。如果你不能确定何时该使用深度链接功能，请参考Uber API Design Guidelines（Uber API设计规范）。

IOS

For deep linking on iOS, first check to see if the Uber app is installed on the user’s device. If the Uber app is installed, launch the Uber app; if the Uber app is not installed, launch Uber’s mobile site (m.uber.com). Note, [Universal Links], introduced in iOS 9, are not supported at the moment.

在iOS上实现深度链接时，首先应当检查用户的设备上是否已经安装了Uber客户端。如果已安装Uber客户端，直接打开；如果没有安装，则加载Uber的移动版网站（m.uber.com）。注意：现在暂时还不支持在iOS9当中新加入的【Universal Links】功能。

CHECKING IF THE UBER APP IS INSTALLED 检查Uber客户端是否已经安装

Enabling uber:// Schema Links in Your App (REQUIRED on iOS 9+)
在你的应用中启用uber:// Schema地址（需要iOS9+系统）

You'll need needs to whitelist Uber in your app's Info.plist file with an `LSApplicationQueriesSchemes` entry similar to the following. This will authorize calls to `canOpenURL` to check for deep links with the `uber://` schema.
你需要在你的应用程序的Info.plist文件中加入一段如下的LSApplicationQueriesSchemes。这样会授权调用canOpenURL来检查uber:// Schema类型的深度链接。

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>uber</string>
</array>
```

To determine if the Uber app is installed on iOS, use the following call:
要确认是否Uber客户端已经在iOS上安装，使用如下调用方法：

```
if ([[UIApplication sharedApplication] canOpenURL:[NSURL
URLWithString:@"uber://"]]) {
    // Do something awesome - the app is installed! Launch
App.
}
else {
    // No Uber app! Open Mobile Website.
}
```

LAUNCHING THE UBER APP
启动Uber客户端

To launch the Uber app, use the following call (at a minimum):
要启动Uber客户端，使用如下调用（最少参数）：

`uber://?action=setPickup&pickup=my_location`

You can add additional query parameters as listed below. Note that there is no need to add an additional `setDropoff` action. Please refer to the examples below as needed.
你可以添加如下所示的额外查询参数。注意，不要加入一个额外的setDropoff动作。如果有需要的话请参考如下的示例。

Query Parameters
查询参数

Passing parameters can make the Uber experience even more seamless for your users. For example, dropoff location parameters can be used to automatically pass the user’s destination information over to the driver.
通过传递参数，可以让你的用户的Uber体验更加无缝。例如，dropoff位置参数可以用来自动将用户的终点信息传递给司机。

Parameters used in linking must be URL-encoded with %20 for spaces. Note that the keys for `pickup` and `dropoff` are the same. `nickname` and `formatted_address` may be omitted if they are the same.
所有链接当中的参数都需要使用URL编码（空格编码为%20）。注意，pickup和dropoff当中的参数键值是一致的。nickname和formatted_address如果一致的话可以省略。

参数名	类型	描述
`client_id`	`string`	Client identifier of the requesting application 发起访问的客户端标识符
`product_id`	`string`	The product to populate in pickup 发起接载的Uber产品
`pickup[latitude]`	`float`	The latitude coordinate for pickup 接载地点的纬度坐标
`pickup[longitude]`	`float`	The longitude coordinate for pickup 接载地点的经度坐标
`pickup[nickname]`	`string`	A URL-encoded `string` of the pickup location name URL编码后的接载位置名字
`pickup[formatted_address]`	`string`	URL-encoded address URL编码后的地址
`pickup=my_location`	(none)	User’s current location 用户当前地址

参数名	类型	描述
dropoff[latitude]	float	The latitude coordinate for dropoff 终点的纬度坐标
dropoff[longitude]	float	The longitude coordinate for dropoff 终点的经度坐标
dropoff[nickname]	string	A URL-encoded string of the dropoff location name URL编码后的终点位置名字
dropoff[formatted_address]	string	URL-encoded address URL编码后的地址

Example 示例
Set pickup and dropoff locations with latitude & longitude pairs, nicknames and formatted addresses. Select Uber product by passing `product_id` returned from the Products endpoint.
设置接载和终点的经纬度、名称和格式化地址。通过传递从产品（Products）终端接口返回的product_id参数，选择Uber产品。

```
uber://?client_id=YOUR_CLIENT_ID&action=setPickup&pickup[latitude]=37.775818&pickup[longitude]=-122.418028&pickup[nickname]=UberHQ&pickup[formatted_address]=1455%20Market%20St%2C%20San%20Francisco%2C%20CA%2094103&dropoff[latitude]=37.802374&dropoff[longitude]=-122.405818&dropoff[nickname]=Coit%20Tower&dropoff[formatted_address]=1%20Telegraph%20Hill%20Blvd%2C%20San%20Francisco%2C%20CA%2094133&product_id=a1111c8c-c720-46c3-8534-2fcdd730040d
```

APPLYING PROMO CODES
应用促销代码

To launch the Uber app and then apply a promo code for the logged in user on iOS:
在iOS上开启Uber应用，并为已登录用户应用促销代码：

```
  uber://?action=applypromo&promo=mypromo
```

The `applypromo` action only takes a single parameter, the promo code value.
applypromo动作只有一个单独的参数，即促销代码。

Query Parameters 查询参数
The single parameter, `promo`, is required is the actual promo code value to apply.
唯一的参数，promo，需要是一个真实的可应用的促销代码。

参数名	类型	描述
`promo`	`string`	The promo code 促销代码

ANDROID

For deep linking on Android, first check to see if the Uber app is installed on the user’s device. If the Uber app is installed, launch the Uber app; if the Uber app is not installed, launch Uber’s mobile site (m.uber.com). Here’s an example:

要在Android上使用深度链接，同样需要首先检查Uber客户端是否已经被安装在用户的设备上。如果Uber客户端已经安装，则之前启动Uber客户端；如果Uber客户端尚未安装，则加载Uber移动版网站（m.uber.com）。下面是示例：

```
try {
  PackageManager pm = context.getPackageManager();
  pm.getPackageInfo("com.ubercab",
PackageManager.GET_ACTIVITIES);
  String uri =
      "uber://?
action=setPickup&pickup=my_location&client_id=YOUR_CLIENT_ID"
;
  Intent intent = new Intent(Intent.ACTION_VIEW);
  intent.setData(Uri.parse(uri));
  startActivity(intent);
} catch (PackageManager.NameNotFoundException e) {
  // No Uber app! Open mobile website.
  String url = "https://m.uber.com/sign-up?
client_id=YOUR_CLIENT_ID";
  Intent i = new Intent(Intent.ACTION_VIEW);
  i.setData(Uri.parse(url));
  startActivity(i);
}
```

Android supports the same URL and query parameters as iOS, so please refer to the App Launch section for iOS for more information and examples.

Android支持与iOS一样的URL地址和查询参数，所以更多信息和示例请直接参考iOS的App Launch（应用启动）部分。

WINDOWS PHONE

The Uber app for Windows Phone also supports the same URL and query parameters as iOS and Android, so please refer to the App Launch section for iOS for more information and examples.
Windows Phone上的Uber客户端同样支持与iOS和Android一致的URL地址和查询参数，所以更多信息和示例请直接参考iOS的App Launch（应用启动）部分。

At this time, there is no way to determine if a user has installed the Uber app on the Windows Phone.
现在暂时还没有办法判断一个用户是否在他的Windows手机上安装了Uber客户端。

MOBILE WEB
移动网页

For scenarios where the user does not have the Uber app installed on their device, deep link to the Uber mobile site (m.uber.com)
在用户没有在设备上安装Uber原生客户端的场景下，需要深度链接到Uber移动网站上（m.uber.com）。

LAUNCHING THE UBER MOBILE SITE 加载Uber移动网站

To launch the Uber mobile site (m.uber.com), use the following call and pass your client ID to allow the user to sign up for Uber:

要加载Uber移动网站（m.uber.com），使用如下调用并且传递你的客户端标识符来允许用户注册Uber：

```
GET https://m.uber.com/sign-up?client_id=YOUR_CLIENT_ID
```

Query Parameters 查询参数

Passing parameters can make the Uber experience even more seamless for your users. To make the signup process more seamless, pass additional query parameters about the user. Also, dropoff location parameters can be used to automatically pass the user's destination information over to the driver.

通过传递参数，可以让你的用户的Uber体验更加无缝。要让用户注册流程更加无缝，请传递与用户相关的额外查询参数。同样的，dropoff位置参数可以用来自动传递用户的终点信息给司机。

Parameters used in linking must be URL-encoded with %20 for spaces.
所有链接当中的参数都需要使用URL编码（空格编码为%20）。

参数名	类型	描述
`client_id`	`string`	Client identifier of the requesting application. Required to correctly attribute new signups originating from your app. 发起访问的客户端标识符，用来准确统计从你的app中新注册的用户。
`first_name` (optional)	`string`	First name of the user. Recommended to provide seamless signup experience. 用户姓名中的名。为了更无缝的注册体验，推荐使用。
`last_name` (optional)	`string`	Last name of the user. Recommended to provide seamless signup experience. 用户姓名中的姓。为了更无缝的注册体验，推荐使用。

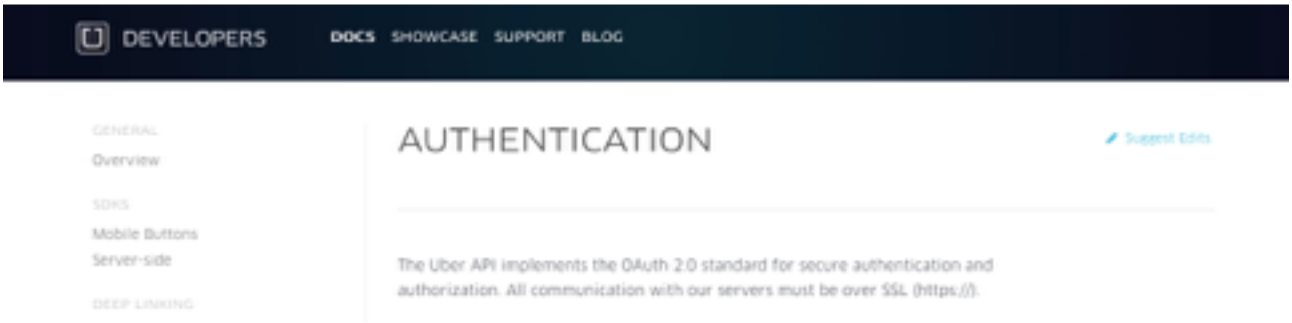
参数名	类型	描述
`email` (optional)	`string`	User's email address. Recommended to provide seamless signup experience. 用户的email地址。为了更无缝的注册体验，推荐使用。
`country_code` (optional)	`string`	2 Letter ISO Country code. 2位字母的标准国家代码
`mobile_country_code` (optional)	`string`	Country code for the mobile phone number. 移动电话号码对应的国家区号
`mobile_phone` (optional)	`string`	Mobile phone number. 移动电话号码
`zipcode` (optional)	`string`	User's zip code. 用户的邮编
`device_id` (optional)	`string`	Unique device identifier. 唯一设备标识符
`android_device_id` (optional)	`string`	IMEI for Android devices. Android设备的IMEI码
`google_aid` (optional)	`string`	Google advertising identifier (For Android devices). Google广告标识符（Android设备）
`ios_aid` (optional)	`string`	iOS advertising identifier (For iOS devices). iOS广告标识符（iOS设备）
`windows_aid` (optional)	`string`	Windows advertising identifier (For Windows devices). Windows广告标识符（Windows设备）

参数名	类型	描述
`product_id` (optional)	`string`	Product option selected by the user. 用户选择的产品选项
`pickup_latitude` (optional)	`float`	Latitude component of pickup location. 接载地址的纬度参数
`pickup_longitude` (optional)	`float`	Longitude component of pickup location. 接载地址的经度参数
`pickup_nickname` (optional)	`string`	Nickname to show for pickup location. 接载地址的名字
`pickup_address` (optional)	`string`	Formatted pickup address to show to user. 展示给用户所需要的格式化接载地址
`dropoff_latitude` (optional)	`float`	Latitude component of dropoff location. 终点地址的纬度参数
`dropoff_longitude` (optional)	`float`	Longitude component of dropoff location. 终点地址的经度参数
`dropoff_nickname` (optional)	`string`	Nickname to show for dropoff location. 终点地址的名字
`dropoff_address` (optional)	`string`	Formatted dropoff address to show to user 展示给用户所需要的格式化终点地址

Example 示例

Set pickup and dropoff locations with latitude & longitude pairs, nicknames and formatted addresses. Select Uber product by passing `product_id` returned from the Products endpoint.
设置接载和终点的经纬度、名称和格式化地址。通过传递从产品（Products）终端接口返回的product_id参数，选择Uber产品。

```
https://m.uber.com/sign-up?client_id=YOUR_CLIENT_ID
&first_name=myFirstName&last_name=myLastName&email=test@examp
le.com
&country_code=us&mobile_country_code=
%2B1&mobile_phone=123-456-7890
&zipcode=94111&product_id=a1111c8c-
c720-46c3-8534-2fcdd730040d&pickup_latitude=37.775818
&pickup_longitude=-122.418028&pickup_nickname=Uber%20HQ
&pickup_address=1455%20Market%20St%2C%20San%20Francisco%2C
%20CA%2094103
&dropoff_latitude=37.802374&dropoff_longitude=-122.405818
&dropoff_nickname=Coit%20Tower
&dropoff_address=1%20Telegraph%20Hill%20Blvd%2C%20San
%20Francisco%2C%20CA%2094133
```



AUTHENTICATION

The Uber API implements the OAuth 2.0 standard for secure authentication and authorization. All communication with our servers must be over SSL (https://).

AUTHENTICATION
鉴权

The Uber API implements the OAuth 2.0 standard for secure authentication and authorization. All communication with our servers must be over SSL (https://).
Uber的API使用OAuth 2.0标准来保证安全的鉴权与授权过程。所有与我们的服务器通信的过程必须使用SSL（即https://）。

AUTHENTICATION OR AUTHORIZATION?
鉴权还是授权?

Before you begin, you should determine what level of access your application needs. Many applications will only use the Products, Price Estimates, and Time Estimates endpoints. For these, you only need to use the server_token to access resources via the API Token Authentication.
在你开始之前，你应当确定你的应用程序需要哪一个级别的访问权限。许多应用程序其实只需要用到产品（Products）、价格估计（Price Estimates）和时间估计（Time Estimates）等终端接口。如果仅仅是这些接口，你只需要使用server_token来通过API Token鉴权访问所需资源。

If your application will access resources on behalf of an Uber user, such as with the Me and User Activity endpoints, you will need to follow the three-legged OAuth 2.0 flow in order to obtain an access_token.
如果你的应用程序将会调用一些有关Uber用户的信息，比如我（Me）、用户活动（User Activity）等终端接口，则需要遵循以下的三段式OAuth 2.0流程来获取一个access_token。

API TOKEN AUTHENTICATION
API Token鉴权

In order to access protected resources you can pass your API Token, called a server_token. This must be passed as an Authorization header with type Token.
为了访问受保护的某些资源信息，你需要使用一个叫server_token的API Token，它必须以Token类型在授权的消息头中传递。

```
curl -H 'Authorization: Token YOUR_SERVER_TOKEN' \
'https://api.uber.com/v1/products?
latitude=37.7759792&longitude=-122.41823'
```

OAUTH 2.0

OAuth 2.0 is a specification outlined in RFC 6749 that allows third-party services to make requests on behalf of a user without accessing passwords and other sensitive information. If you are unfamiliar with OAuth 2.0, check out Aaron Parecki's "OAuth 2 Simplified" guide. We recommend that you use a pre-built library to perform grant and token exchanges. Python, Ruby, or NodeJS.
OAuth 2.0是在RFC 6749当中写明的规范，它允许第三方服务在不访问密码或者其它敏感信息的情况下，代表用户发起请求。如果你对OAuth 2.0不熟悉，请查阅Aaron Parecki的文档《OAuth 2 Simplified》。我们推荐你使用一个预先构建好的库，比如Python、Ruby或者NodeJS，来实现Token的获取和交换。

STEP ONE: AUTHORIZE
第一步：授权

The first step of the flow is to redirect a user to the following URL, with the appropriate query parameters for your application. The response will be an HTML web form where the resource owner is authenticated with the Uber API and is prompted to approve or deny your request for OAuth 2.0 permissions. The scope requested (or the application defaults) will be displayed.
流程中的第一步是将用户重定向到如下URL，带上你的应用程序所需的合适的查询参数。响应结果会以HTML web格式呈现，在这里信息所有者会对Uber API进行鉴权，通过或者是拒绝你的OAuth 2.0权限请求。页面上将会展示请求访问的范围（或者程序的默认设置）。

```
GET https://login.uber.com/oauth/v2/authorize
```

参数名	描述
`response_type`	OAuth 2.0 response type. `code` is the only acceptable input at this time. OAuth 2.0响应类型。现在唯一支持的输入是`code`。
`client_id`	The client ID of your application. 你的应用程序的客户端标识符。

参数名	描述
`scope` (optional)	Space delimited list of grant scopes you would like to have permission to access on behalf of the user. 这里用来界定你希望得到用户授权后，可以代表用户访问的范围。
`state` (optional)	State which will be passed back to you to prevent tampering. 为了防止被干预，State会被回传给你。
`redirect_uri` (optional)	The URI we will redirect back to after an authorization by the resource owner. The base of the URI must match the `redirect_uri` used during the registration of your application. 在资源信息所有者进行授权后，我们会重定向到的URI。该URI的基础部分必须匹配你在注册你的应用程序时使用的redirect_uri。

STEP TWO: RECEIVE REDIRECT
第二步：接收重定向

Once the Uber user authenticates and authorizes your app, Uber will issue an HTTP 302 redirect to the redirect_uri passed in. On that redirect, you will receive an authorization code, which is single use and expires in 10 minutes.
一旦Uber用户鉴权并授权给你的应用程序后，Uber会下发一个HTTP 302链接重定向到传进来的redirect_uri。在这个重定向链接里，你会收到一个鉴权码，该鉴权码只可使用一次，且有效期只有10分钟。

```
GET https://your-redirect-uri/?code=AUTHORIZATION_CODE
```

STEP THREE: GET AN ACCESS TOKEN
第三步：获取Access Token

Exchange this authorization code for an access_token, which will allow you to make requests on behalf of a user. The access_token expires in 30 days. The grant_type may be authorization_code or refresh_token.

使用上面得到的授权码来交换一个access_token，它可以允许你代表用户发起请求。access_token的有效期为30天。grant_type参数可以是authorization_token或者是refresh_token。

```
curl -F 'client_secret=YOUR_CLIENT_SECRET' \
    -F 'client_id=YOUR_CLIENT_ID' \
    -F 'grant_type=authorization_code' \
    -F 'redirect_uri=YOUR_REDIRECT_URI' \
    -F 'code=AUTHORIZATION_CODE' \
    https://login.uber.com/oauth/v2/token
```

响应：

```
{
    "access_token": "EE1IDxytP04tJ767GbjH7ED9PpGmYvL",
    "token_type": "Bearer",
    "expires_in": 2592000,
    "refresh_token": "Zx8fJ8qdSRRseIVlsGgtgQ4wnZBehr",
    "scope": "profile history"
}
```

STEP FOUR: USE BEARER TOKEN

第四步：使用不记名Token

Pass the access_token returned in the response in the Authorization header with the type Bearer to make requests on behalf of a user.

在代表用户发起请求时，使用不记名（Bearer）类型将access_token放在授权的消息头中。

```
curl -H 'Authorization: Bearer YOUR_ACCESS_TOKEN' 'https://
api.uber.com/v1/products?
latitude=37.7759792&longitude=-122.41823'
```

STEP FIVE: REFRESHING TOKENS

第五步：刷新Token

When the user's access_token has expired, you may obtain a fresh access_token by exchanging the refresh_token that is associated with the access_token.

当用户的access_token过期的时候，你可以使用该access_token对应的refresh_token来换取一个全新的access_token。

```
curl -F 'client_secret=YOUR_CLIENT_SECRET' \
    -F 'client_id=YOUR_CLIENT_ID' \
    -F 'grant_type=refresh_token' \
    -F 'redirect_uri=YOUR_REDIRECT_URI' \
    -F 'refresh_token=REFRESH_TOKEN' \
    https://login.uber.com/oauth/v2/token
```

STEP SIX: REVOKING TOKENS
第六步：禁用Token

To revoke a user's access to the Uber API via your application, you may query our revoke endpoint with the user's access_token. This action is equivalent to a user visiting their user profile and clicking ‘Disconnect'.
要在你的应用程序中禁用一个用户对于Uber API的访问，你需要使用用户的access_token来调用我们的禁用（revoke）终端接口。这个动作与用户访问他们的用户主页并点击“解除绑定”效果是一致的。

```
curl -F 'client_secret=YOUR_CLIENT_SECRET' \
    -F 'client_id=YOUR_CLIENT_ID' \
    -F 'token=ACCESS_TOKEN' \
    https://login.uber.com/oauth/revoke
```

REFERENCE
参考文档

APPLICATION PARAMETERS
应用程序参数

After you've registered your developer account and application, you'll able to access the following parameters for your application on the Applications page.
在你注册了你的开发者账号和应用程序之后，你可以在应用（Applications）页面，拿到你应用程序对应的以下参数。

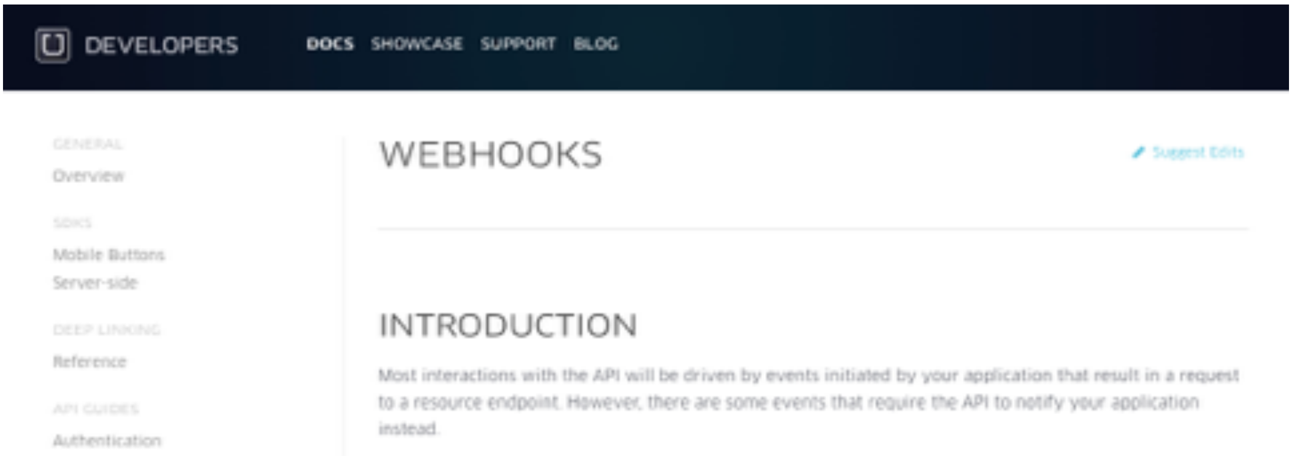
参数名	描述
`client_id`	A 32 character string (public) 一个32位的字符串（公开）

参数名	描述
`client_secret`	A 40 character string. DO NOT SHARE. This should not be available on any public facing server or web site. If you have been compromised or shared this token accidentally please reset your client_secret via our web interface. This will require updating your application to use the newly issued `client_secret`. 一个40位的字符串。**不要分享给任何人。**这不应该出现在任何对外公开的服务器或者是网站上。如果你意外泄露了该参数，请通过我们的网站重置你的client_secret。这将会需要你的应用程序更新到全新下发的client_secret。
`server_token`	A 40 character string. DO NOT SHARE. This must be used for requests that are not issued on behalf of a User. 一个40位的字符串。**不要分享给任何人。**当你不是代表用户发起请求时，必须使用这个Token。

AUTHENTICATION ERRORS
鉴权错误

参数名	描述
`invalid_request`	Required parameters were not provided. 缺少必须提供的参数。
`invalid_client`	The client ID or secret provided is invalid. 提供的客户端标识符和秘钥无效。
`invalid_grant`	Valid grant types are `authorization_code` and `refresh_token` 有效的分发类型只有authorization_code和refresh_token。

参数名	描述
`invalid_scope`	The `scope` parameter provided is not a valid subset of scopes. 访问范围（scope）参数无效。
`server_error`	The server returned an unknown error. 服务器返回了未知错误。
`temporarily_unavailable`	The endpoint is temporarily unable to respond. 终端接口暂时无法响应。



WEBHOOKS

INTRODUCTION

Most interactions with the API will be driven by events initiated by your application that result in a request to a resource endpoint. However, there are some events that require the API to notify your application instead.

WEBHOOKS
网络回调

INTRODUCTION
简介

Most interactions with the API will be driven by events initiated by your application that result in a request to a resource endpoint. However, there are some events that require the API to notify your application instead.

大部分与API的交互都是由你的应用程序发起的事件驱动，最终发起到终端接口的请求。但是，也有一些时候，需要API反过来通知你的应用程序。

Web hooks solves this problem by allowing you to specify a URL that the Uber API will make requests to. The Uber platform will then notify your application with relevant events. Edit the WEBHOOK URL parameter of your application and provide a URL your application can listen on to get started.
网络回调通过让Uber API来访问一个由你指定的URL地址，解决了这个问题。届时，Uber平台将会因为相关的事件对你进行通知。通过编辑你的应用程序的WEBHOOK URL参数，提供一个你的应用程序可以监听的URL地址来开始吧。

JSON BODY POST PARAMETERS
POST请求JSON类型消息体相关参数

参数名	类型	描述
`event_id`	`string`	Unique event identifier, which can be used to ensure that events are only digested once. 唯一事件标识符，通过它可以确保事件只被处理一次。
`event_time`	`integer`	Unix timestamp of the time the event occurred. 事件发生的时间对应的Unix时间戳。
`event_type`	`string`	The type of event that occurred. 发生的时间的类型。
`meta`	`object`	The object containing additional information that is specific to the `event_type`. 这个对象含有对应不同event_type的额外信息。
`resource_href`	`string`	Contains the URL to GET to request the entire resource. 包含一个可以使用GET来请求全部资源信息的URL地址。

EXAMPLE POST

请求示例

```
{
    "event_id": "3a3f3da4-14ac-4056-bbf2-d0b9cdcb0777",
    "event_time": 1427343990,
    "event_type": "requests.status_changed",
    "meta": {
        "resource_id": "2a2f3da4",
        "status": "in_progress"
    },
    "resource_href": "https://api.uber.com/v1/requests/2a2f3da4"
}
```

EVENTS
事件

REQUESTS.STATUS_CHANGED

For all Requests made by your application on behalf of Uber riders, we will make a request to your WEBHOOK URL whenever its status changes. This can help you notify the user or change the state of your app to reflect a status change without continuously polling the `/v1/requests` endpoint.

对于所有通过你的应用程序代表用户发起的用车请求，每当它的状态发生变化时，我们都会发起一个对你的网络回调（WEBHOOK）URL地址的请求。这将有助于你实时通知你的用户或者变更你的应用程序的状态，而无需持续调用 /v1/requests终端接口。

META OBJECT
对应的META对象

参数名	类型	描述
`resource_id`	`string`	Unique identifier of the Request which has had a status change. 发生状态变化的用车请求对应的唯一标识符。

参数名	类型	描述
`status`	`string`	The status of the Request indicating state that just changed. 表明状态刚刚被改变的用车请求对应的状态。
`resource_type`	`string`	The type of resource: `request` 资源类型：request

EXAMPLE POST
请求示例

```
{
    "event_id": "3a3f3da4-14ac-4056-bbf2-d0b9cdcb0777",
    "event_time": 1427343990,
    "event_type": "requests.status_changed",
    "meta": {
        "resource_id": "2a2f3da4",
        "resource_type": "request",
        "status": "in_progress"
    },
    "resource_href": "https://api.uber.com/v1/requests/2a2f3da4"
}
```

REQUESTS.RECEIPT_READY

For all Requests made by your application on behalf of Uber riders we will make a request to your WEBHOOK URL whenever a Request Receipt is available. This will allow you to show your user the details of their fare and how much they were charged as soon as their Receipt is available. If the rider cancels after the grace period, and they are charged, a receipt will still be available showing that charge.
对于所有通过你的应用程序代表用户发起的用车请求，每当可以获取到用车收据时，我们都会发起一个对你的网络回调（WEBHOOK）URL的请求。这将允许你在可获取用车收据的第一时间，将用户的资费详情和他们被收取的费用展示给他们。

To get access to a Request Receipt resource, the user must have authorized your application to have access to the `request_receipt` scope.
要访问“用车收据”信息，用户必须已经授权你的应用程序访问request_receipt域。

META OBJECT
对应的META对象

参数名	类型	描述
`resource_id`	`string`	Unique identifier of the Request Receipt which has had a status change. 有状态变化的“用车收据”请求对应的唯一标识符。
`status`	`string`	The status of the Request Receipt indicating the state that just changed. 表明状态刚刚被改变的“用车收据”请求对应的状态。
`resource_type`	`string`	The type of resource: `request_receipt` 资源类型：request_receipt

EXAMPLE POST
请求示例

```
{
    "event_id": "2a2f3da4-14ac-4056-bbf2-d0b9cdcb0777",
    "event_time": 1427343990,
    "event_type": "requests.receipt_ready",
    "meta": {
        "resource_id": "d0b9cdc",
        "resource_type": "request_receipt",
        "status": "ready"
    },
    "resource_href": "https://api.uber.com/v1/requests/
d0b9cdc/receipt"
}
```

REFERENCE
参考文档

SANDBOX
沙盒

When making Requests in the sandbox environment, we will also emit webhook events for simulated trips. The `requests.status_changed` event will be emitted every time the status of a simulated Request changes. The `requests.receipt_ready` event will be emitted once a Request’s status changes to completed.
当在沙盒环境中发起请求时，我们同样会为了仿真流程触发网络回调（Webhook）事件。事件requests.status_changed会在每次仿真请求状态有改变时触发；事件requests.receipt_ready会在任何一个请求状态改为“已完成”时触发。

HEADERS
消息头

The Uber API will insert specialized headers for all requests made to your WEBHOOK URL to help your application utilize them appropriately.
Uber API会在所有对你的网络回调URL地址发起的请求里加上特殊的消息头，以帮助你的应用程序正确合理的应用这些请求。

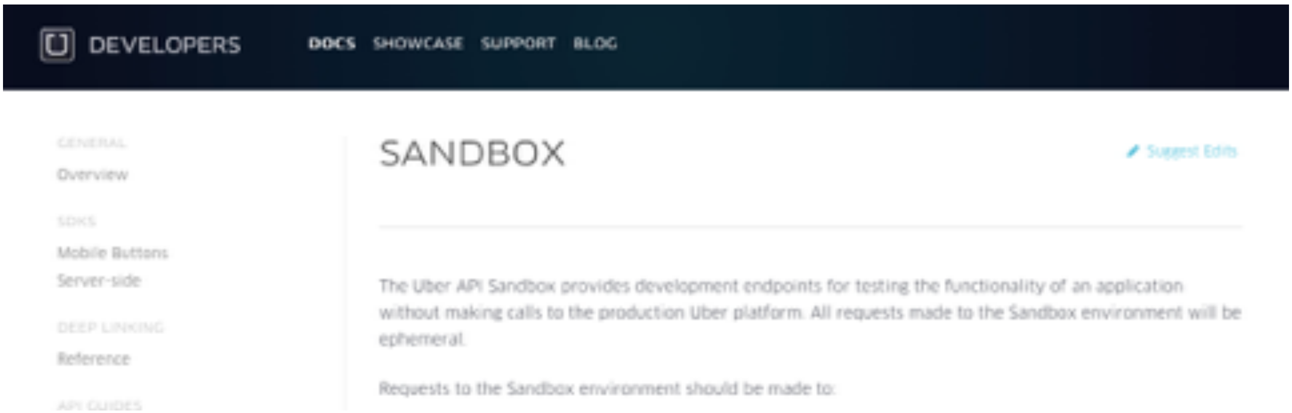
消息头	描述
`X-Environment`	Indicates if this request is coming from the `production` or `sandbox` API. 标明该请求是来自于正式线上环境或是沙箱的API。

RETRY ATTEMPTS
重试

If for some reason the Uber API cannot reach the WEBHOOK URL you specified either due to networking issues or application issues on your end, the webhooks service will retry to make a request.
如果因为某些网络或者你的应用程序侧的原因，Uber API未能成功访问你指定的网络回调URL地址，网络回调服务会重新尝试发起一个请求。

We have implemented an exponential back-off algorithm with a back-off multiplier of 30 seconds which will make up to 7 attempts. This means we will attempt to request your WEBHOOK URL up to 7 times across roughly 1 hour. Initial try at 0 seconds, followed by 30 seconds, then 60 seconds, then 120, etc.

我们实现了一个以30秒为乘数的指数级增长延时算法，最多发起7次请求。这意味着我们会在大约1小时时间内尝试7次访问你的网络回调URL地址。第一次是0秒，第二次是30秒，第三次是60秒，然后是120秒等等。



SANDBOX

The Uber API Sandbox provides development endpoints for testing the functionality of an application without making calls to the production Uber platform. All requests made to the Sandbox environment will be ephemeral.

Requests to the Sandbox environment should be made to:

SANDBOX
沙盒

The Uber API Sandbox provides development endpoints for testing the functionality of an application without making calls to the production Uber platform. All requests made to the Sandbox environment will be ephemeral.
Requests to the Sandbox environment should be made to:
Uber API沙盒为测试应用程序功能提供了开发终端接口，开发者无需访问正式线上环境的Uber平台。所有对沙盒环境发起的请求都是临时的，这些请求需要对以下地址发起：

```
https://sandbox-api.uber.com/<version>
```

At this time, other than the Request endpoints, all other endpoints are proxied straight to the production environment. This is because Request is currently the only write enabled endpoint that is part of the Uber API.
现在，除了Request类的终端接口，所有其它终端接口均会被直接代理到正式线上环境，因为Request终端接口是Uber API里唯一支持写入的部分。

All OAuth users and their tokens that are available in the production Uber API environment will also be valid to make requests to the Sandbox API.

所有在Uber API正式线上环境可使用的OAuth用户账户和他们的token均可以有效地向沙盒API发起请求。

REQUEST
用车请求

Currently the sandbox does not change states automatically the way a real Request in production would, so this endpoint gives the ability to walk an application through the different states of a Request.
现在沙盒还不会像正式线上环境当中的真实用车请求一样自动改变状态，所以该终端接口提供了让应用程序挨个调试一个用车请求各个不同状态的能力。

This endpoint effectively just modifies the status of an ongoing sandbox Request.
该终端接口实际上只修改正在沙盒中进行的用车请求的状态。

RESOURCE
资源

```
PUT /v1/sandbox/requests/{request_id}
```

Accepts a JSON body indicating how you would like to manipulate the status of a Request.
接受一个JSON类型的消息体，它用来表明你想如何控制一个用车请求的状态。

JSON BODY PARAMETERS
JSON类型消息体相关参数

参数名	类型	描述
`status`	`string`	The value to change a Request’s status to. 要将一个用车请求的状态修改至的新状态的值。

```
Content-Type: application/json
```

`{"status": "accepted"}`

RESPONSE
响应结果

```
Status-code: 204 Success
状态代码：204 Success
```

EXAMPLE CURL
CURL语句示例

```
  curl -X "PUT" "https://sandbox-api.uber.com/v1/sandbox/requests/" \
  -H "Authorization: Bearer " \
  -d "{\"status\":\"accepted\"}"
```

PUT REQUEST STATUSES
PUT用车请求状态

状态	描述
`processing`	The Request is matching to the most efficient available driver. 该用车请求正在匹配最合理的空闲司机。
`accepted`	The Request has been accepted by a driver and is “en route” to the start location (i.e. `start_latitude` and `start_longitude`). 该用车请求已经被某位司机接受，且“正在途中”前往接载起始位置（即起始纬度start_latitude和起始经度start_longitude）
`arriving`	The driver has arrived or will be shortly. 司机已经或者即将到达。

状态	描述
`in_progress`	The Request is “en route” from the start location to the end location. 该用车请求“正在途中”从起始点前往终点。
`driver_canceled`	The Request has been canceled by the driver. 该用车请求被司机取消。
`completed`	Request has been completed by the driver. 该用车请求已经被司机完成。

OTHER REQUEST STATUSES
其它用车请求状态

The following statuses cannot be set via PUT requests. Instead, they are set via other API calls as described in the table.
以下状态无法通过PUT请求修改得到，它们需要通过下表中描述的API修改得到。

状态	描述
`no_drivers_available`	The Request was unfulfilled because no drivers were available. Use the Product Types sandbox endpoint to create a Request with this status. 因为没有空闲司机，所以该用车请求无法被满足。请使用产品类型（Product Types）沙盒终端接口来发起一个此状态的用车请求。
`rider_canceled`	The Request canceled by rider. Issue a `DELETE` command to give a Request this status. 该用车请求被乘客取消。请发起删除（DELETE）命令将一个用车请求置于此状态。

PRODUCT TYPES
产品类型

Used to simulate the possible responses the Request endpoint will return when requesting a particular product, such as surge pricing, against the Sandbox.
与沙盒相反，被用来模拟用车请求接口在发起一个特定产品的请求时可能返回的响应结果，比如“提价”（surge pricing）。

RESOURCE
资源

```
PUT /v1/sandbox/products/{product_id}
```

Accepts a JSON body indicating what you would like the `surge_multiplier` to be when making a Request to a particular Product.
接受一个JSON类型的消息体，它用来表明你在发起一个特定产品的用车请求时希望设置的提价倍数。

JSON BODY PARAMETERS
JSON类型消息体相关参数

参数名	类型	描述
`surge_multiplier`	`float`	The surge multiple a Product should have when making a Request in the Sandbox. A multiplier greater than or equal to `2.0` will require the two stage confirmation screen. 一个产品在沙盒中发起用车请求时可能会收到的提价倍数。当倍数大于或者等于2.0时，会需要在屏幕上进行两步确认。
`drivers_available`	`boolean`	If `False`, attempts to make a Request in the Sandbox will return a `no_drivers_available` error 如果为False，在沙盒中发起的用车请求将会返回一个no_drivers_available错误。

```
Content-Type: application/json
```

```
{
  "surge_multiplier": 2.2,
  "drivers_available": true
}
```

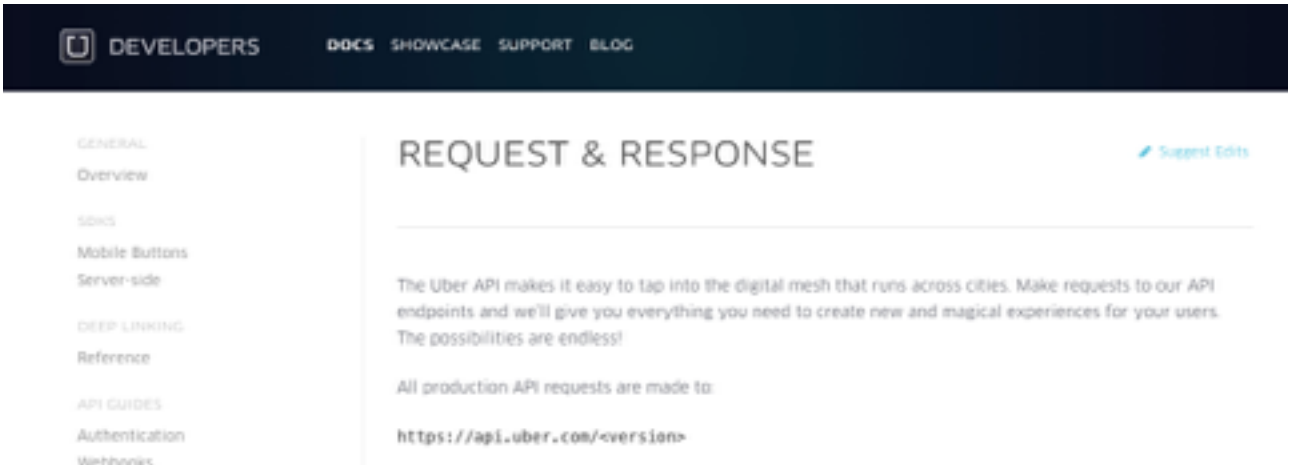
```
{"drivers_available": false}
```

RESPONSE
响应结果

```
Status-code: 204 Success
状态码: 204 Success
```

EXAMPLE CURL
示例CURL语句

```
  curl -X "PUT" "https://sandbox-api.uber.com/v1/sandbox/
products/" \
  -H "Authorization: Bearer " \
  -d "{\"surge_multiplier\":2.2, \"drivers_available\":true}"
```



REQUEST & RESPONSE

The Uber API makes it easy to tap into the digital mesh that runs across cities. Make requests to our API endpoints and we'll give you everything you need to create new and magical experiences for your users. The possibilities are endless!

All production API requests are made to:

```
https://api.uber.com/<version>
```

REQUEST & RESPONSE
请求与响应

The Uber API makes it easy to tap into the digital mesh that runs across cities. Make requests to our API endpoints and we’ll give you everything you need to create new and magical experiences for your users. The possibilities are endless!

Uber API可以让你更加便捷地接入这个在无数个城市中流畅运行的数字网络。通过向我们的API终端接口发起请求，我们会给你所需的一切，让你给你的用户创造全新而神奇的体验。

All production API requests are made to:
所有正式线上环境的API请求都应该发往：

```
https://api.uber.com/<version>
```

The current version of the API is v1. Backwards incompatible changes will result in a version bump. Some of our API endpoints require OAuth 2.0 credentials. Please see the Authentication & Authorization guide to get started.
当前的API的版本号为v1，向下不兼容的改变将会导致版本变化。我们的API当中的一些需要OAuth 2.0证书。请查阅鉴权与授权（Authentication & Authorization）来开始开发吧。

There is also a Development Sandbox to use when developing and testing applications, with requests being made to:
另外，在开发和测试应用程序的时候，我们还提供了一个开发用沙盒环境。要使用沙盒API，所有的请求应该发往：

```
https://sandbox-api.uber.com/<version>
```

COMMON TYPES
常规类型

Date and time will be expressed as an integer which represents the number of seconds since the Unix epoch in UTC.
日期和时间会以整型（integer）来表达，它以世界协调时（UTC）表达了自Unix新纪元以来的秒数（Unix时间戳）。

Latitude and longitude will be expressed as floats, with the precision on both the request and response up to six decimal places.
经纬度会以浮点数来表达，在请求和响应当中最高都会精确到6位小数。

CORS (CROSS-ORIGIN RESOURCE SHARING)
CORS 跨域资源共享

The Uber API supports CORS for communicating from Javascript. You will need to set an Origin URI when creating your application to allow for CORS to whitelisted for your domain. Please note that your server_token is still considered sensitive and it is your responsibility to protect it.
Uber API支持跨域资源共享（CORS），用以完成从Javascript发起的通信。你需要在创建你的应用程序时设定一个Origin URI，以便CORS将你的域加入白名单。请注意，你的server_token仍然是非常敏感的数据，你需要注意妥善保存。

```
var xhr = new XMLHttpRequest();
xhr.setRequestHeader("Authorization", "Token
YOUR_SERVER_TOKEN");
xhr.open('GET', 'https://api.uber.com/v1/products?
latitude=37.7759792&longitude=-122.41823');
```

HTTP STATUS CODES
HTTP状态码

Status Code	Description
200	OK. Everything worked as expected. 请求已成功。
201	Created. We will return a `201` after a successful `POST` where a resource was created. 请求已经被实现。当一个资源已经依据一个成功的POST请求而被创建时，我们会返回状态码201。
400	Malformed request. 错误的请求。
401	Unauthorized the request requires user authentication (not logged in) 未经授权的验证，需要用户进行鉴权（未登录）。

Status Code	Description
403	Forbidden. Also used for unauthorized requests such as improper OAuth 2.0 scopes or permissions issues. 禁止访问。也被用于未授权的请求，例如不适当的OAuth 2.0范围和权限问题。
404	Not found. 未找到请求的资源。
406	Unacceptable content type. Client sent an accepts header for a content type which does not exist on the server. Body includes a list of acceptable content types, such as “Unacceptable content type. Request resource as: `application/json`. 不被接受的内容类型。客户端在消息头中请求了一个服务器上不存在的内容类型。在消息体中会给出可被接受的内容类型列表，例如“Unacceptable content type. Request resource as: `application/json`”
409	A conflict needs to be resolved before the request can be made. 存在冲突，在发起请求前必须先解决此冲突。
422	Invalid request. The request body is parse-able however with invalid content or there are issues with a rider’s user account. 无效的请求。请求的消息体格式正确，但是含有语义错误，或者是乘客的用户账户存在问题。
429	Too Many Requests. Rate limited. 请求过多，速率受到限制。
500	Internal Server Error. 服务器内部错误。

ERRORS
错误

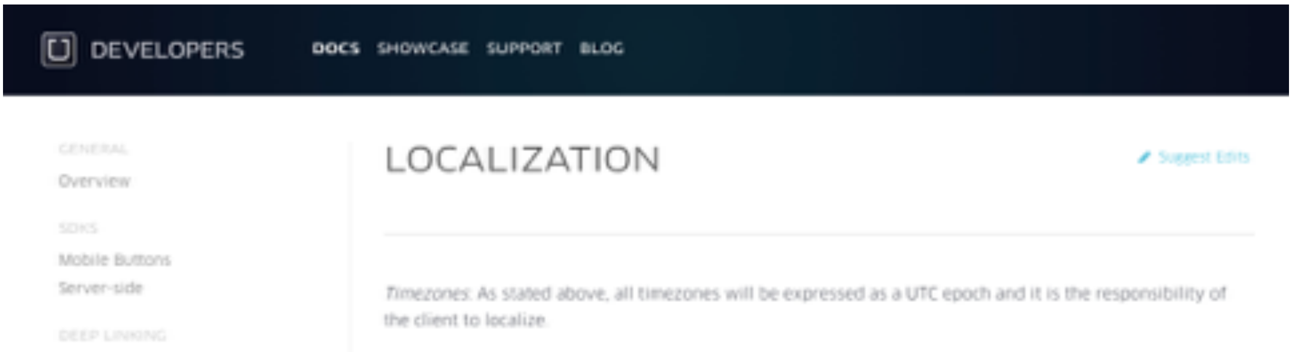
Error responses will have a consistently formed JSON body. The keys may include:
错误响应将会有一个一致的JSON类型消息体，里面的键值可能包括：

键	值
`message`	Human readable message which corresponds to the client error. 有用户可读性的客户端错误相关解释
`code`	Underscored delimited string. 一个下划线分隔的字符串
`fields` (optional)	A hash of field names that have validations. This has a value of an array with member strings that describe the specific validation error. 一组需要校验的字段名组成的数组，数组内包括描述了具体错误原因的字符串。

EXAMPLE
示例

```
Status-Code: 422 Invalid Request
错误码: 422 Invalid Request
```

```
{
  "message": "Invalid user",
  "code": "invalid",
  "fields": {
    "first_name": ["Required"]
  }
}
```



LOCALIZATION

Timezones: As stated above, all timezones will be expressed as a UTC epoch and it is the responsibility of the client to localize.

Timezones: As stated above, all timezones will be expressed as a UTC epoch and it is the responsibility of the client to localize.

时区：如前所述，所有的时区时间都将会以UTC时间戳来表示，客户端需要自行按当前时区进行本地化。

Currency: Currency will be rendered in the local currency of a given latitude & longitude pair. A fare estimate in San Francisco will be given in dollars, while a fare estimate in Paris will be displayed in euros. We also provide the ISO 4217 currency code for your own conversions.
货币：货币将会根据所提供的经纬度确定的地点自动转换成当地币种。在旧金山提供的价格估计将会以美元的形式提供，而在巴黎则会是欧元。我们同时还为你自己转换提供了 ISO 4217 标准货币代码。

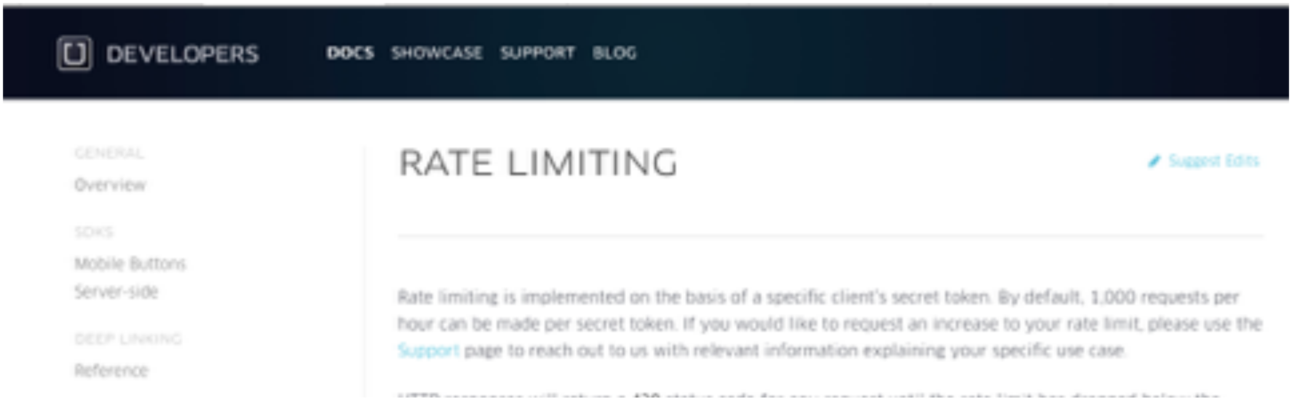
Translations: Uber API respects an `Accept-Language` header given by the client for languages for which we support translations. We support Section 14.4 of RFC 2616. We will return a `Content-Language` response header with the content’s localization.
翻译：Uber API 根据客户端给予的 Accept-Language 消息头来给予翻译支持，我们支持 Section 14.4 of RFC 2616 。我们将会返回一个带有 Content-Language 消息头的响应，该响应内的内容为相应的本地化内容。

Uber API, Uber’s mobile website and mobile applications support a variety of languages and locales specified via standard Accepts-Language headers. We continue to add more languages as we add more locales. Currently, the following locales are supported:
Uber 的 API 、移动网站和移动应用支持了一批语种和地点，均可由标准的 Accepts-Language 消息头定制。随着提供服务的地点的增加，我们正在继续添加更多的语种支持。现在，可以支持以下的地点：

代码	描述
`ar_SA`	Arabic (Saudi Arabia) 阿拉伯语（沙特阿拉伯）
`de_DE`	German (Germany) 德语（德国）
`en_US`	English (United States) 英语（美国）

代码	描述
`fr_FR`	French (France) 法语（法国）
`it_IT`	Italian (Italy) 意大利语（意大利）
`ja_JP`	Japanese (Japan) 日语（日本）
`ko_KR`	Korean (Korea) 韩语（韩国）
`ms_MY`	Malay (Malaysia) 马来语（马来西亚）
`nl_NL`	Dutch (Netherlands) 荷兰语（荷兰）
`pt_BR`	Portuguese (Brazil) 葡萄牙语（巴西）
`ru_RU`	Russian (Russia) 俄语（俄罗斯）
`sv_SE`	Swedish (Sweden) 瑞典语（瑞典）
`th_TH`	Thai (Thailand) 泰语（泰国）
`tl_PH`	Tagalog (Philippines) 菲律宾语（菲律宾）
`zh_CN`	Chinese (China) 中文（中国）

代码	描述
`zh_TW`	Chinese (Taiwan) 中文（中国台湾）



RATE LIMITING

Rate limiting is implemented on the basis of a specific client's secret token. By default, 1,000 requests per hour can be made per secret token. If you would like to request an increase to your rate limit, please use the Support page to reach out to us with relevant information explaining your specific use case.

HTTP responses will return a 429 status code for any request until the rate limit has dropped below the

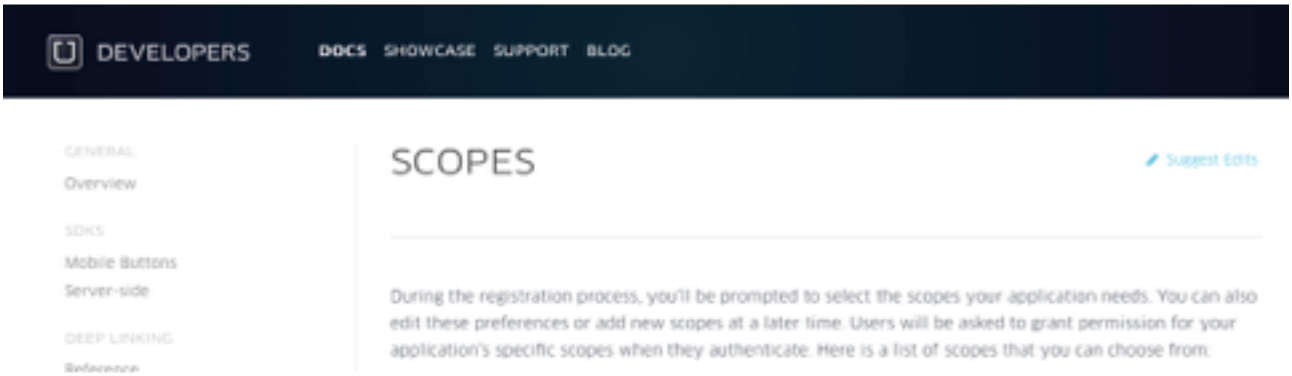
速率限制

Rate limiting is implemented on the basis of a specific client's secret token. By default, 1,000 requests per hour can be made per secret token. If you would like to request an increase to your rate limit, please use the Support page to reach out to us with relevant information explaining your specific use case.
速率限制根据特定客户端的 token 而有所不同，默认情况下每个token在一个小时内能发起1000个请求。如果你想要增加你的上限，请在支持 Support 页面联系我们，提供相关信息解释你的使用情况。

HTTP responses will return a `429` status code for any request until the rate limit has dropped below the required threshold.
在速率降低到限制一下之前，所有的 HTTP 响应都会返回429代码。

Furthermore, we will supply `X-Rate-Limit-*` headers for all requests containing the current state of the secret token's rate limit.
另外，我们给所有的请求提供了 `X-Rate-Limit-*` 信息头，包含有当前的 token 对应的速率限制的状态。

信息头	描述
`X-Rate-Limit-Limit`	Total number of requests possible. 可发起的请求的总数。
`X-Rate-Limit-Remaining`	Number of requests left in the rate limit window. 在速率限制窗口内剩余的可发起请求数。
`X-Rate-Limit-Reset`	Timestamp when the rate limit will reset. 速率限制将会被重置的时间戳。



SCOPES

During the registration process, you'll be prompted to select the scopes your application needs. You can also edit these preferences or add new scopes at a later time. Users will be asked to grant permission for your application's specific scopes when they authenticate. Here is a list of scopes that you can choose from.

SCOPES
域

During the registration process, you’ll be prompted to select the scopes your application needs. You can also edit these preferences or add new scopes at a later time. Users will be asked to grant permission for your application’s specific scopes when they authenticate. Here is a list of scopes that you can choose from:
在注册的时候，你将要选择你的应用程序所需要访问的域。你也可以在之后修改或者添加新域。用户在鉴权的时候将会被询问是否授予你的应用程序所有你想得到授权的域。以下是所有你可以选择的域的列表：

域	描述
`profile`	Access the basic profile information on a user's Uber account including their first name, email address, and profile picture. 访问用户的Uber账户的基本资料信息，包括他们的名、email地址、用户头像。
`history`	Pull trip data including times, product type, and city information of a user's historical pickups and drop-offs. 拉取行程信息，包括时间、产品类型和用户过往的接载地点以及目的地的城市信息。
`history_lite`	Same as history but without city information. 和history一样，但是没有城市信息。
`request`	Make requests for Uber Products on behalf of users. 代表用户发起Uber产品的请求。
`request_receipt`	Get receipt details for Requests made by application. 获取由此应用程序发起的用车请求对应的详细收据。

VERSIONING
版本更新

The Uber API is constantly being worked on to add features, make improvements, and fix bugs. This means that you should expect changes to be introduced and documented. Whenever we make a significant change to an endpoint, we will increase the version number used in the path of the resource being requested.
我们一直在为Uber API增加、改进功能，修复bug。这意味着你随时会看到我们介绍并且在文档中记录最新的变化。当我们对某个终端接口做出很显著的修改时，我们就会提高相应的版本号。

However, there are some changes or additions that are considered backwards-compatible and your applications should be flexible enough to handle them. These include:

但是，有一些改变或者添加会被认为是向下兼容的，你的应用程序应当能够灵活地处理这些情况，包括：

- Adding new endpoints to the API
- 在API中增加新的终端接口
- Adding new attributes to the response of an existing endpoint
- 在已有终端接口的响应中增加新的参数
- Changing the order of attributes of responses (JSON by definition is an object of unordered key/value pairs)
- 改变响应中参数的顺序（按定义，JSON应当是由一组无顺序的键值对组成的对象）

DESIGN GUIDELINES
设计规范

Deliver a better user experience by applying the Uber design guidelines.
遵循Uber设计规范，给予用户更佳的使用体验。

ASSETS AND GUIDELINES
资源与规范

For guidelines specific to using the API endpoints and generally integrating Uber into your app, please use the following guidelines and assets.
要查找使用API接口或者是在你的应用中整合Uber的规范，请参考以下规范和资源。

- Rides API Design Guidelines
- Rides API Flow Diagrams
- Rides API Design Assets
- Ride Reminders API UX Guidelines

UBER PRESS KIT
Uber媒体资源

For Uber-approved marketing assets including our logotype, app screenshots and general photography, download the Uber Press Kit.

下载Uber Press Kit以获得Uber许可的正式市场推广资源，包括我们的标志、应用截图、照片等等。

CHINA

Supporting your app in China

OVERVIEW

This tutorial will summarize everything you need to know to ensure your App supports riders in China - including information about how our internal systems partition users/apps and subtle differences in our API endpoints.

CHINA
中国

Supporting your app in China
在中国支持你的应用程序

OVERVIEW
概览

This tutorial will summarize everything you need to know to ensure your App supports riders in China - including information about how our internal systems partition users/apps and subtle differences in our API endpoints.
该教程会总结你的应用在支持中国的乘客的过程当中所需要了解的一切，包括我们内部系统分区的用户、应用程序和我们API终端接口的细微差别。

API REFERENCE

Visiting our API documentation in China will automatically give you China specific implementation details for the Uber APIs. You will notice slight variations between US/Worldwide Apps and China Apps. All of these variations are also summarized below for your convenience.
访问我们的站点 API documentation in China（中国API文档），它会自动调整内容为中国区特殊的Uber API实施细节。你会注意到在美国、全球区域的应用

和中国区应用间存在微小的差异。为了你使用的方便，所有的这些差异被总结在了下方。

1. Create your app
1. 创建你的应用程序

Create your app in the Uber Developer Dashboard. The dashboard exists in two different regions depending on the targeting of your app:
在Uber开发者面板中创建你的应用程序，根据你的应用程序的目标用户不同，开发者面板出现在两个不同的区域。

用例	开发者网站
为中国区之外的乘客提供服务的应用	https://developer.uber.com/
为中国区的乘客提供服务的应用	https://developer.uber.com.cn/

App metadata, such as app name, client ID, server token, is stored independently in different regions. So if you register an app in the US for example, it will not exist in China, and vice versa.
应用的元数据，例如应用名、客户端ID、服务器token，都分别在不同区域存储。所以你如果在美国注册一个应用程序，它将不会出现在中国，反之亦然。

2. Authentication and tokens
2. 鉴权与token

Generate your tokens as described on the Authentication page. The hostname of the login server you use will depend on the targeting of your app.
依据鉴权页面的提示，可以生成你的token。登录服务器的主机名将取决于你的应用的目标市场。

用例	登录服务器主机名
为中国区之外的乘客提供服务的应用	https://login.uber.com/
为中国区的乘客提供服务的应用	https://login.uber.com.cn/

Note: tokens are not replicated between service regions.
注意：在不同的服务区域，token不是复用的。

3. API calls
3. API调用

Make API requests as described in the endpoints documentation. The hostname of the API server depends on the targeting of your app.
依据终端接口文档中的描述发起API请求，API服务器的主机名将取决于你的应用的目标市场。

用例	登录服务器主机名
为中国区之外的乘客提供服务的应用	https://api.uber.com/
为中国区的乘客提供服务的应用	https://api.uber.com.cn/

The following endpoints have dependencies on specific regional services.
以下终端接口依赖于具体提供服务的地区。

- /v1/requests. The region should be selected based on pickup/dropoff location of the ride. For example, a ride request in Beijing, China should be made with the api.uber.com.cn and a ride request for London, England should be made with the api.uber.com. After a ride, any additional calls related to the ride are restricted to the region in which it was created.
- /v1/requests：地区的选择需要基于行程的起点和终点的位置。例如，一个在中国北京发起的行程请求需要访问 api.uber.com.cn ；而一个在英国伦敦发起的请求则需要访问 api.uber.com 。在行程结束之后，任何该行程相关的附加接口调用，都需要按该次行程创建的地区来进行。
- /v1.x/history. The region should be selected based on the location of the user.
- /v1.x/history：地区的选择需要基于用户的地理位置。
- /v1/estimates/time. The region should be selected based on the pickup and dropoff latitude/longitude.
- /v1/estimates/time：地区的选择需要基于起点和终点的经纬度。

Endpoints not mentioned above do not have any dependencies on particular regions.
上面没有提到的终端接口不依赖于具体的地区。

Error Handling

错误处理

There may be cases in which an API call is made using the wrong regional service. In this case, a 302 response code will be returned.

有些情况下，API调用使用了错误的地区，此时，会返回302响应代码。

For example, consider a request a request to retrieve user history for a China user from the US service:

例如，发起一个请求给美国区服务器调用一个中国用户的行程历史：

```
curl -X "GET" "https://api.uber.com/v1.2/history?offset=0&limit=10" \
    -H "Authorization: Bearer ACCESS_TOKEN_FOR_CHINA_RIDER" \
    -H "Content-Type: application/json"
```

The response will look like the following:

响应会如下所示：

```
HTTP/1.1 302 retry_request_on_uri
Location: https://api.uber.com.cn/v1.2/history

"meta":{},"errors":[{"status":302,"code":"retry_request_on_uri","title":"Retry request on given redirect URI."}]}
```

The response contains a 302 redirect, which means you should retry your API call in the other region.

响应会包含一个302重定向，意味着你需要重新使用另一个地区参数发起你的API调用。

UPDATES AND CHANGES

更新与变动

We are working hard on platform improvements to make the Uber API easier to use internationally. Please keep updated by subscribing to the Uber Developer Blog.

我们正在努力改进整个平台，使Uber API能够更便捷地提供国际化服务。请订阅Uber Developer Blog（Uber开发者博客）以了解最新信息

FAQ

HOW LONG DOES THE DISPATCH SYSTEM WAIT BEFORE RETURNING NO_DRIVERS_AVAILABLE?

The dispatch system tries to find a driver for 2 minutes before returning no_drivers_available.

I'M HAVING TROUBLE WITH OAUTH. WHAT SHOULD I DO?

FAQ

HOW LONG DOES THE DISPATCH SYSTEM WAIT BEFORE RETURNING NO_DRIVERS_AVAILABLE?
派单系统在等待多长时间后会返回无空闲司机（NO_DRIVERS_AVAILABLE）响应?

The dispatch system tries to find a driver for 2 minutes before returning no_drivers_available.
派单系统将持续2分钟时间尝试寻找空闲司机，如果2分钟仍未找到，则会返回no_drivers_available。

I’M HAVING TROUBLE WITH OAUTH. WHAT SHOULD I DO?
我在使用OAUTH时碰到了麻烦，我该怎么做?

We recommend working through our Authentication section and our auth tutorials to get the basics of OAuth worked out. If you are still having issues, contact us here.
我们推进您阅读鉴权（Authentication）部分文档和相关的教程来掌握使用OAuth的基本知识。如果你仍然有疑问的话，请在此联系我们。

HOW DO I GET ACCESS TO WHITELISTED ENDPOINTS?
我如何才能访问拥有白名单机制的终端接口?

At this time, some Uber API endpoints are in beta. The partners that are receiving early access to these endpoints are those that have already built and launched interesting Uber API integrations without them. Feel free to share you integrations with us and we’ll consider adding you to our beta tests. As always, keep an eye out for updates via Twitter @Uber_API!
一些Uber的API终端接口暂时还是处于beta阶段，能收到这些测试终端接口的早期访问权限的合作伙伴，都是已经利用现有Uber的API完成了有趣的集成的。请随时与我们分享您的整合方案，我们会考虑将您加入到我们的beta测试环节。同样的，还请通过Twitter @Uber_API随时关注我们的最新动态。

ARE THERE ANY EXAMPLES OF THE API IN ACTION?
有现有API的调用示例吗?

Absolutely. You can check out our showcase page to see a wide range of examples and case-studies. If you build a killer app, we might even showcase your app too!
当然，你可以在我们的showcase页面找到广泛的示例和案例分析。如果你制造了一个“杀手级”应用，我们也有可能在这里展示你的应用！

HOW CAN I RECEIVE UPDATES ON THE UBER API?
我能如何收取Uber API的最新更新呢?

Follow us on Twitter @Uber_API. We’ll be posting updates there!
请在Twitter上关注我们@Uber_API，我们会在此账号发布更新消息。

WHAT IS THE UBER API AFFILIATE PROGRAM?
Uber合作伙伴计划是什么?

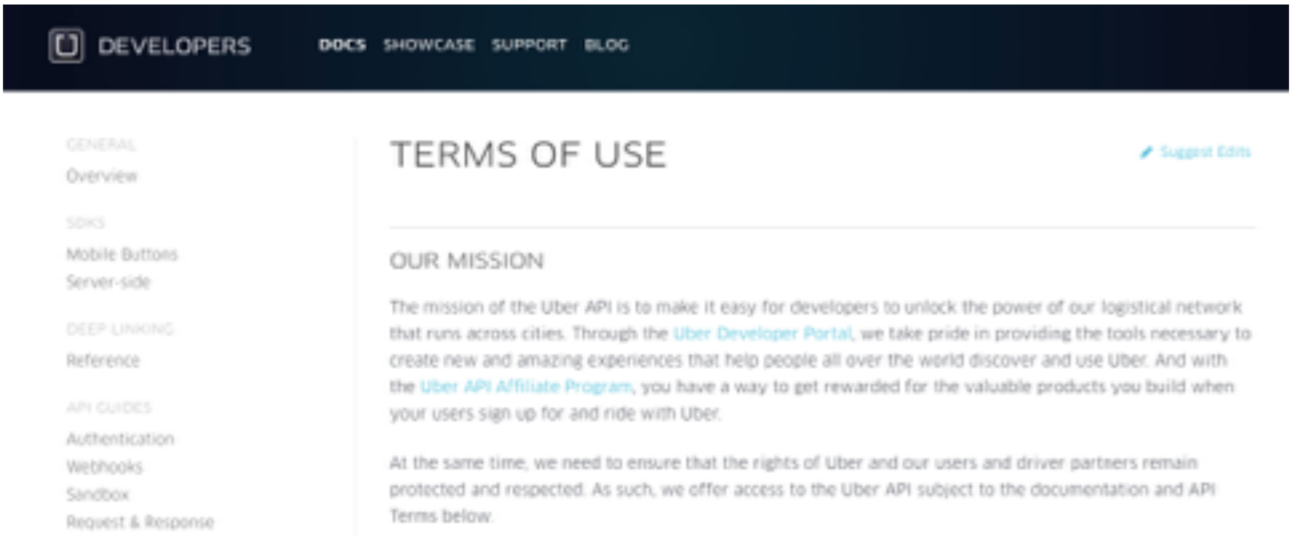
The Uber API Affiliate Program is a program for developers who are using the Uber API to earn Cash Rewards for every new rider that signs up through their app. If you are a US-based developer you can earn $5 USD for every new US-based Uber user you refer. Learn more on our Earn Rewards page.
Uber API合作伙伴计划是一项给予使用Uber API的开发者现金奖励的计划，每当有一位司机通过他们的应用注册，开发者就能收到对应奖励。如果你是一位在美国的开发者，每当有居住在美国的Uber用户通过你的推荐注册，你将收到$5。进入我们的赢取奖励页面可以查看更多详情。

CAN I USE THE WORD "UBER" IN MY APP'S NAME?
我可以在我的应用程序名字里加上Uber吗?

You may use Uber Marks to refer to the availability of Uber API features or functionality in connection with your Services, both in your Services or in a description of your Services on an app store. For example, you may include the Uber logo in your Services to indicate where the user can obtain a fare quote or request a ride, or you may mention in the app store description of your Services that Uber functionality is included. However, you may not name your Services in any way that suggests that Uber may own or sponsor the Services. For example, you should not use “Uber” as part of the name of your company or your Services. For more information, please review our Terms of Use.
你可以在你的服务页面或者是应用商店的描述当中，使用Uber商标来表示你的服务中能够支持Uber API的相关特性或功能。例如，你可以在你的服务页面加入Uber的标志，来表示用户可以得到行程价格估算或者是请求一次行程等服务，或者你也可以在应用商店的描述中提及已经支持Uber相关功能。但是，你不可以使用任何会引起误解的名称，导致人们以为这是Uber自有或者是投资的服务。例如，在你的公司或者应用名称当中，不可以使用Uber。更多相关信息，请参考使用条款（Terms of Use）页面。

DO YOU HAVE AN AFFILIATE PROGRAM FOR DRIVERS?
你们有针对司机的合作伙伴计划吗?

No, at the moment the affiliate program is only for riders.
不，暂时只有针对乘客的伙伴计划。



TERMS OF USE

OUR MISSION

The mission of the Uber API is to make it easy for developers to unlock the power of our logistical network that runs across cities. Through the Uber Developer Portal, we take pride in providing the tools necessary to create new and amazing experiences that help people all over the world discover and use Uber. And with the Uber API Affiliate Program, you have a way to get rewarded for the valuable products you build when your users sign up for and ride with Uber.

At the same time, we need to ensure that the rights of Uber and our users and driver partners remain protected and respected. As such, we offer access to the Uber API subject to the documentation and API Terms below.

TERMS OF USE
使用条款

OUR MISSION
我们的使命

The mission of the Uber API is to make it easy for developers to unlock the power of our logistical network that runs across cities. Through the Uber Developer Portal, we take pride in providing the tools necessary to create new and amazing experiences that help people all over the world discover and use Uber. And with the Uber API Affiliate Program, you have a way to get rewarded for the valuable products you build when your users sign up for and ride with Uber.

Uber API的使命是为了使开发者更便捷地解锁我们遍布各大城市的发达交通网络。通过 Uber Developer Portal（Uber开发者门户），以帮助全世界用户发现和使用Uber为目标，我们为创造全新的、令人惊叹的用户体验提供了必要的工具，并为之感到骄傲。

At the same time, we need to ensure that the rights of Uber and our users and driver partners remain protected and respected. As such, we offer access to the Uber API subject to the documentation and API Terms below.

We are looking for partners that align with the following principles:

同时，我们也需要保证Uber的相应权利，并且我们的用户、司机也应该受到保护和尊重。因此，我们需要在遵循以下文档和API使用条款的条件下，提供访问Uber API的选项。我们依据以下原则挑选我们的合作伙伴：

Be a good citizen on Uber's network:
在Uber网络中做一个优秀的公民：

- Respecting the rights and privacy of our users and driver partners is very important to us. Please do not spam or otherwise mislead our users or driver partners in any way.
- 尊重我们的用户和司机的权利和隐私对我们而言是非常重要的，请不要发送垃圾信息或者以任何形式误导我们的用户和司机。
- User and driver partner cancellations happen infrequently; we'd like to keep it that way. Please do not take actions that intentionally increase cancellation rates.
- 某些极少数情况下，用户和司机才会取消行程，我们很乐意保持现状。请勿使用某些方法有意地提高取消率。
- We work tirelessly to perfect the user pickup so that users can easily find their vehicles and get moving to their final destination. Please do not misrepresent a user's pickup location or destination.
- 我们不知疲倦地工作是为了让用户的行程更加完美，使用户可以更加便捷地找到他们的车辆并且迅速地到达他们的目的地。请不要歪曲用户的接载地点或者目的地。
- Many of our driver partners rely on Uber to make a living. Focus on ways to help improve our platform by directing legitimate traffic to our driver partners.
- 许多我们的司机伙伴以Uber为生，请专注于能够有效帮助平台的正确方法，例如永远给司机正确合法的导航，
- Be creative and make Uber an even more magical experience for people. Help them get things done more easily and cost effectively with Uber.
- 请保持创造力，让Uber可以给人带来更加神奇的体验，帮助人们更好、更快捷、更低成本地使用Uber。

Be a strong, trustworthy partner to Uber. Please do not:
要成为一个强大的、值得信赖的Uber合作伙伴，请不要：

- Compete with Uber or try to drive traffic away from Uber.
- 与Uber进行竞争或者是试图对Uber进行分流。
- Resell Uber's services.

- 转售Uber服务。
- Aggregate Uber with competitors.
- 同时集成Uber和其竞争对手。
- Store or aggregate Uber's data, except as expressly permitted by Uber.
- 存储或者汇集Uber的数据，除非得到Uber的明确许可。
- Slander or disparage Uber.
- 诽谤或者贬低Uber。

INTRODUCTION
概述

Uber Technologies, Inc. and its affiliates ("Uber," "we," or "us") are excited to provide development access to the Uber API and our applications, websites and technology platform ("Uber Services"). We are seeking the best in developer partners to grow our collective products and services. However, we need to balance offering an open platform for innovative development with ensuring the protection of Uber, the Uber Services and our users and partners. These Uber API Terms of Use ("API Terms") strive for that balance and set forth our policies regarding permitted use of the Uber API along with some necessary restrictions and legal protections.

Uber科技有限公司，或者其代称（“Uber”、“我们”）十分兴奋于能够提供Uber API、Uber应用程序、网站和科技平台（“Uber服务”）作为开发平台。我们正在寻找最好的开发者合作伙伴来共同促进我们的产品和服务的发展。然而，我们也需要在为创新提供开放平台和有效保证对Uber、Uber服务、Uber用户和伙伴的保护中间取得平衡。这些Uber API使用条款（“API条款”）就是为了达到这个平衡，并且提出在遵循必要的限制和法律保护的条件下可以允许使用Uber API的政策。

The API Terms will evolve as our business and platform expands and as creative developers dream up new applications for the Uber API, so check back often and make sure that you are familiar with the most current version. We encourage you to contact us if you have questions or suggestions about the Uber API or these API Terms.

API使用条款会随着我们的事业和平台的不断扩张和富有创造力的开发者为Uber API不断开发新应用的过程不断优化，所以请随时查阅以保证您对于最新的条款非常熟悉。如果您有任何关于Uber API或者API使用条款的疑问或者建议，我们非常鼓励您联系我们（contact us）。

1. ACCESS & REGISTRATION

1. 访问与注册

A. Acceptance

A. 许可

All use of (i) the Uber API, (ii) any data accessed or obtained via the Uber API ("Uber Data"), and (iii) Uber API-related documentation, software, and materials provided or made available to you by Uber (collectively, the "Uber API Materials") is subject to and must comply with these API Terms. By using the Uber API Materials, you acknowledge that you have read, and agree to abide by, these API Terms. Please stop using the Uber API if you determine that you are unable to comply with the current or any future version of the API Terms. We may modify or update the Uber API from time to time. While we can provide no guarantee, we will try to ensure that future versions of the Uber API are backwards compatible to at least the previous version to the extent reasonably possible.

所有对于（i）Uber API，（ii）任何通过Uber API获得的数据（Uber数据），（iii）Uber API相关的文档、软件以及任何Uber提供或者开放给您的相关材料（统称为“Uber API材料”）的使用都应当且必须遵循这些API使用条款。通过使用Uber API材料，表示您确认您已阅读并且同意遵守这些API使用条款。如果您认为您无法遵循当前或者以后某个版本的API使用条款，请停止使用Uber API。我们有可能随时对Uber API进行修改和更新，同时，我们无法保证，但我们会尽量使将来的Uber API版本至少向后兼容到至少前一版本合理可能的范围内。

We reserve the right to revoke access to Uber API without notice if your use of the Uber API violates any of these API Terms or if we otherwise object to your use of the Uber API.

我们保留不另行通知的情况下取消您对Uber API的使用资格的权利，如果您对于Uber API的使用违反了任何相关API使用条款，或者我们会通过其它方式终止您使用Uber API。

B. Your Account and Registration

B. 您的账户与注册

To access the Uber API, you will first need to register as an Uber API Developer. You will also need to register each of your applications that use the Uber API with us. All user and application registration information can be found here. After registering, we will issue you with a Developer ID and secret code (called the client_id and client_secret). Your Developer ID and code is required for all calls to the Uber API by your application.

为了使用Uber API，您必须首先注册为Uber API开发者。同时，您还必须将您任何使用了Uber API的应用程序向我们进行注册。所有的用户与应用程序注册信息都可以在这里（here）找到。在注册之后，我们会分发给您一个开发者ID和一个密码（被称为client_id和client_secret）。在每次您的应用程序向Uber API发起调用时，都会需要您的开发者ID和密码。

C. Fees
C. 费用

Access to and use of the Uber API Materials and the Uber Services available in connection with the Uber API are currently provided at no charge. However, we reserve the right to charge for access and/or use of the Uber API Materials and/or Uber Services in the future at our discretion, including, without limitation, rated pricing and/or differentiated pricing for business users. We will provide you with notice in the event we decide to start charging for use.

现阶段访问和使用Uber API材料、通过Uber API的连接提供的Uber服务是免费的。然而，我们保留在未来自行决定对访问、使用Uber API材料和/或Uber服务进行收费的权力，包括但不限于针对企业用户的额定定价和/或差异化定价。我们会在决定开始对使用收费时向您发出通知。

2. USE POLICIES
2. 使用政策

A. Acceptable Use
A. 合理使用

You may use the Uber API Materials in connection with your applications, products or services that are registered with us (your "Services") to make available certain various features and functionality of the Uber Services via the Uber API. However, you may not require payment of any kind by users for using the Uber API Materials, or any aspect of the Uber Services that may be available or accessible in connection with the Uber API.

您可以通过向我们进行了注册的应用程序、产品、服务（您的“服务”）使用Uber API材料，来使用Uber API提供特定的某些Uber服务的特性和功能。但是，您不能以任何形式向使用Uber API材料或者是任何通过Uber API提供的Uber服务的用户进行收费。

You may not use the Uber API or Uber Data in any manner that is competitive to Uber or the Uber Services, including, without limitation, in connection with any application, website or other product or service that also includes, features, endorses, or otherwise supports in any way a third party that provides services competitive to Uber's products and services, as determined in our sole discretion.
您不能以任何形式使用Uber API或者Uber数据与Uber或者Uber服务进行竞争，包括但不限定于，连接到包含、提供功能、宣传或者以任何形式支持第三方与任何与Uber的产品和服务提供竞争关系的服务的应用程序、网站或者服务业务。竞争关系将由我们自行确定。

Your use of the Uber API may be subject to certain limitations on access, Uber Data requests, and use as set forth in the API Terms, on the Uber Developer Portal, or as otherwise provided to you. If we believe that you have attempted to exceed or circumvent these limitations, your ability to use the Uber API may be temporarily or permanently blocked. We may monitor your use of the Uber API and Uber Data to improve the Uber Services and to ensure compliance with these API Terms.
您对于Uber API的使用需要遵循特定接口访问、Uber数据请求和由API使用条款确定的限制，这些限制将通过Uber开发者门户（Uber Developer Portal）或者其它方式提供给您。如果我们确认您试图超越或者规避这些限制，您使用Uber API的权限将被临时或者永久封锁。我们可能会监控您对于Uber API和Uber数据的使用，以优化Uber服务和确保API使用条款被有效遵守。

You agree that Uber may collect certain use data and information related to your use of the Uber API and Uber Data in connection with your Services ("Usage Data"), and that Uber may use such Usage Data for any business purpose, internal or external, including, without limitation, providing enhancements to the Uber API or Uber Services, providing developer of user support, or otherwise.
您同意Uber可能会收集您在通过您的服务使用Uber API时产生的特定使用数据和信息（“使用数据”），并且Uber可能出于内部或者外部的任何商业目的使用这些使用数据，包括但不限定于提供对Uber API或者Uber服务的优化改进、提供开发者用户支持等等。

You may include advertisements in your Services near your visual implementation of the Uber API (e.g., banner ads above or below), but you must ensure clear separation between Uber content and any advertisements.

Your advertisements should not be displayed in any manner that suggests approval or endorsement by Uber.
您可以在您的服务当中，在靠近您对于Uber API的视觉实现部分展示广告（例如，在上方或者下方的横幅广告），但是您必须确保Uber内容和任何广告区域之间存在明显分隔。

B. Use Restrictions
B. 使用限制

In general, we reserve the sole right to determine whether or not your use of the Uber API or Uber Data is acceptable, and to revoke Uber API access for any Service that we determine is not providing added benefit to Uber users and/or is not in the best interests of Uber or our users.
一般地，我们保留自行决定您对于Uber API或者Uber数据的使用是否可以接受的权利，并且保留停止我们认为对于Uber用户未能提供附加价值和/或不符合Uber或者我们的用户的最佳利益的服务对于Uber API访问的权利。

The following are some, but not all, restrictions applicable to the use of the Uber API and Uber API Materials:
以下是一些，但不是全部，对于使用Uber API和Uber API材料的有效限制：

- You may not use the Uber API, Uber API Materials, or Uber Data in any manner that is competitive to Uber or the Uber Services, including, without limitation, in connection with any application, website or other product or service that also includes, features, endorses, or otherwise supports in any way a third party that provides services competitive to Uber's products and services, in our sole discretion.
- 您不能以任何与Uber或者Uber服务有竞争性的形式使用Uber API、Uber API材料或者Uber数据，包括但不限定于，连接到包含、提供功能、宣传或者以任何形式支持第三方与任何与Uber的产品和服务提供竞争关系的服务的应用程序、网站或者服务业务。竞争关系将由我们自行确定。
- You may not use Uber Data for any purpose other than providing the Service for which you are registered to provide as an Uber API Developer. You may not share Uber Data with any third party except as permitted by us for the use of your Service or by an end user who has affirmatively consented to the sharing of data about such end user. Uber Data about an end user in your possession or control must be deleted by you upon such end user's request or upon such end user's termination or cancellation of the Service.
- 您不能以任何在提供您作为一个Uber API开发者进行了注册的服务之外的目的，使用Uber数据。除非得到我们对于您的服务的许可或者得到终

端用户对于其相关的数据被分享和使用的许可，您不能将Uber数据与任何第三方分享。您所有或者控制的一个终端用户的Uber数据必须在该终端用户进行要求或者终端用户决定终止或者取消服务时，进行删除。

- Your Service must allow the end user of your Service to access such end user's Uber Data that you have collected via the Uber API at the request of such end user. Your Service must provide easily accessible end user support contact information.
- 您的服务必须同意其终端用户对于由您在该终端用户通过Uber API发起请求时收集的其Uber数据进行访问。您的服务必须提供便捷的终端用户联系信息。
- You may not collect, store or aggregate Uber Data in any manner except as permitted by us for the use of your Service. You may not share Uber Data (individually, or in the aggregate) with third parties in any manner.
- 除非得到我们对于您如此使用您的服务的允许，您不能以任何形式收集、存储或者集中Uber数据。您不能以任何形式与第三方共享Uber数据（单独或者集合形式）。
- You may not include or use the Uber API in, or in connection with, any application, website or other product or service that includes content that is disparaging of Uber, libelous or may otherwise be perceived as detrimental or harmful to Uber and its business and reputation, in our sole discretion.
- 您不能在任何或者与其有连接的任何由我们自行确定的包含贬低Uber的、诽谤或者是任何其它可能被认作对Uber和其业务以及声誉有害的内容的应用程序、网站或者是任何其它产品或服务当中，包括或者使用Uber API。
- You may not include or use the Uber API or Uber API Materials in, or in connection with, any application, website or other product or service that includes content that is defamatory, libelous, hateful, violent, obscene, pornographic, unlawful, or otherwise offensive, in our sole discretion.
- 您不能在任何或者与其有连接的任何由我们自行确定的包含诽谤、仇视、暴力、淫秽、色情、非法或者其它冒犯用户的内容的应用程序、网站或者是任何其它产品或服务当中，包括或者使用Uber API。
- You may not use the Uber API or Uber API Materials to distribute any virus, spyware, adware, malware, or other harmful or malicious component.
- 您不能使用Uber API或者Uber API材料来散布任何病毒、间谍软件、广告软件、恶意软件或者任何有害或者怀有恶意的软件。

- You may not use the Uber API or Uber API Materials for any purpose which or might overburden, impair or disrupt the Uber Services or related servers or networks.
- 您不能以任何可能引起超荷负载、削弱或者中断Uber服务或者网络的目的使用Uber API或者Uber API材料。
- You may not use the Uber API or Uber API Materials to distribute unsolicited advertising or promotions, or to send messages, make comments, or initiate any other direct communication or contact with Uber users or partners.
- 您不能使用Uber API或者Uber API材料来散布未经要求的广告或者促销，或者发送消息、发布评论，或者发起任何与Uber用户或者伙伴的直接的交流或者通讯。
- You may not, and may not encourage or authorize others to: (i) remove or alter any proprietary notices or marks on the Uber API or Uber API Materials; (ii) use or access the Uber API or Uber API Materials for purposes of monitoring the availability, performance, or functionality of any of Uber's products and services or for any other benchmarking or competitive purposes; (iii) use or access the Uber API to aggregate, cache, or store geographic location information or other user information accessible via the Uber API; (iv) frame, wrap or otherwise reproduce significant portions of the Uber Services; or (v) reverse engineer, reverse assemble, decompile, modify or attempt to discover any source or object code of the Uber API or any part of the Uber Services.
- 您不能，且不能鼓励或者授权任何他人：（i）去除或者修改任何Uber API或Uber API材料上标注的专利标识；（ii）以监控Uber产品或者服务的可访问性、性能、或功能，或者是任何其它形式的评估或竞争目的，使用或者访问Uber API或Uber API材料；（iii）使用或者访问Uber API来收集、缓存或者存储用户的地理位置信息或其它能够通过Uber API获得的信息；（iv）将Uber服务的显著部分加入框架、重新包装或者以任何其他形式复制；或者（v）反向工程、逆向装配、反编译、修改或者任何尝试查找Uber API或者Uber服务的任何部分的源代码或者目标代码的行为。
- You may not charge end users in any manner for access to or use of the Uber API, Uber API Materials or any services or functionality included in or related to the Uber API or Uber Services. Without limiting the foregoing, you may not sell, rent, lease, sublicense, redistribute or syndicate access to the Uber API or the Uber API Materials, and you may not charge any kind of service, booking or similar fee in connection with any services made available via the Uber Services.

- 您不能以任何形式向用户针对访问或者使用Uber API、Uber API材料或者任何包括在或者与Uber API或Uber服务相关的服务或功能进行收费。在不限制前述规定的前提下，您不能售卖、出租、转授权、重分发或者联合访问Uber API或Uber API材料，并且您不能针对基于与Uber服务相关联可行的任何服务收取任何形式的服务、预定或者类似费用。
- You will at all times use the Uber API and Uber API Materials in accordance with all applicable worldwide laws and regulations and the Uber User Terms, and you may not use the Uber API or Uber API Materials to conduct or facilitate, in any way, activity that is in violation of applicable worldwide laws or regulations or the Uber User Terms.
- 您在使用Uber API或者Uber API材料是，必须一直遵循所有适用的世界范围内的法律和监管规定以及Uber用户条款（https://www.uber.com.cn/legal/chn/terms），并且您不能使用Uber API或者Uber API材料来进行或者以任何形式促进违反适用的世界范围内的法律和监管规定以及Uber用户条款（https://www.uber.com.cn/legal/chn/terms）的行为。
- You must not impose any terms on users of your Service that are inconsistent with these API Terms or the Uber User Terms.
- 您不能强制用户遵循与API使用条款或者Uber用户条款（https://www.uber.com.cn/legal/chn/terms）相矛盾的条款。
- You agree to comply with the design guidelines, including, without limitation, any attribution requirement(s), which design guidelines may be updated by us from time to time, and you understand and agree that Uber has the sole discretion to determine whether your attribution(s) are in accordance with the guidelines.
- 您同意遵循设计规范（https://developer.uber.com/docs/design-guidelines），包括但不限定于，任何属性的要求，其设计原则可能会由我们不时更新，并且您理解和同意Uber有权自行决定您的属性是否遵循设计规范。

In addition to the restrictions above, if you are granted access to the request endpoint, the following additional restrictions apply to your use of the Uber API and Uber API Materials:
在上述限制之外，如果您被授权访问行程请求终端接口，下面的额外限制条款将对您使用Uber API和Uber API材料的行为生效：

- In order to use the functionality of the request endpoint, we may need to present users or driver partners with additional terms or obtain additional consents to share information. You may not, and may not encourage or allow any third party to interfere with, hinder, limit, or modify any such notices or consents required of users or driver partners.

- 为了使用行程请求终端接口的功能，我们可能需要给用户或者司机伙伴展示附加条款或取得附加的共享信息的许可。您不能、也不能鼓励或者允许任何第三方干涉、妨碍、限制或者修改任何此类提示或者需要用户或司机伙伴同意的许可。
- You may not initiate or cancel a request for transportation services without the consent of the end user associated with the trip request.
- 您不能在没有与该次行程请求相关的终端用户的同意的情况下，发起或者取消一次交通服务的请求。
- You may not, and may not encourage or allow any third party to interfere with, hinder, limit, or modify the fare amount, fare multiple (including, without limitation, in the event of a "surge" multiple) or any associated fees.
- 您不能、也不能鼓励或者允许任何第三方干涉、妨碍、限制或者修改价格数值、价格倍数（包括但不限定于“提价”倍数）或任何相关费用。
- You may not modify or misrepresent a user's pickup location or destination in any way.
- 您不能以任何形式修改或者歪曲用户的接载地址或终点地址。
- Use of the request endpoint is subject to certain limitations on access as established by Uber from time to time. If you desire to implement the request endpoint in a manner that would exceed the limitations on access, please contact us.
- 使用行程请求终端接口需要遵循特定的访问限制，这些限制会由Uber不时更新。如果您想要以可能会超过访问限制的方式调用行程请求终端接口，请联系我们（contact us）。

C. User Data & Privacy
C. 用户数据和隐私

We respect the privacy of users and expect you to do the same. The basic rule is this: collect only what you need to provide your Service; collect, use and store it a secure manner; and retain it only so long as you need it.

我们尊重用户的隐私也希望您同样尊重我方的隐私。基本规则如下：只收集您提供服务所必须的数据；用安全的方式收集、使用和储存数据；数据只保留您需要的时间。

You agree to publish and abide by a privacy policy explaining how you collect, store, use, and/or transfer any Personal Data (defined below) via your

Services. You also agree to comply with all privacy and data protection laws applicable to you. If you are located outside of the United States, you must disclose this fact to the user and indicate that by using your Service the user may be allowing or enabling the transfer of Personal Data to a country that may offer less protection with respect to Personal Data.

您同意发布并遵守隐私政策，说明您如何收集、储存、使用和/或通过您的服务传输任何个人数据（定义如下）。您也同意遵守适用于您的所有隐私和数据保护法。如果您位于美国境外，则必须向您的用户披露这一事实，指出通过您的服务，您的用户可能允许或授权传输个人数据到一个对个人数据提供更少保护的国家。

If your use of the Uber API or related Uber Services requires or will likely result in the provision of Personal Data directly to Uber, you agree to obtain all necessary consents and authorizations from the applicable users to provide such Personal Data to Uber. Uber will treat Personal Data obtained from you through your use of the Uber API in accordance with its posted Privacy Policy.

如果您对于Uber API或相关Uber服务的使用需要或者可能导致直接向Uber提供个人数据，您同意从向Uber提供这些个人数据的适格用户处获取所有必要的准许与授权。Uber会依照已发布的隐私政策来处理在您使用Uber API过程中从您处获得的个人数据。

Unless otherwise permitted by applicable law and/or agreement with the user, if a user revokes access to their Uber account, you must ensure that all Personal Data pertaining to that user is deleted from your Services and related networks, systems and servers as soon as reasonably possible. If you stop using the Uber API altogether or if your Uber API access is revoked, you must delete all Personal Data in the same way.

除非适用的法律和/或与用户的协议另有许可，如果一个用户取消了对他的Uber账户的访问权限，您必须保证在合理时间内从您的服务和相关网络、系统和服务器中尽快删除所有与该用户有关的个人数据。如果您完全停止使用Uber API 或者被取消对Uber API的访问权限，您必须以同样的方式删除所有的个人数据。

For the purposes of these API Terms, "Personal Data" means information that may be used, either alone or together with other information, to identify an individual user, including, without limitation, a user's name, address, telephone number, username, email address, city and country, picture or other similar information.

依据本API使用协议的目的，“个人数据”是指单独使用或和其他信息一起使用的用以识别用户个体的信息，包括但不限于用户的名字、地址、电话号码、用户名、邮箱地址、城市和国家、照片或其他类似信息。

3. RIGHT & OWNERSHIP

3. 权利和所有权

A. Use of Uber Marks

*A. Uber*商标的使用

The rights granted in these API Terms do not include any general right to use the Uber name or any Uber trademarks, service marks or logos (the "Uber Marks") with respect to your Services. Subject to your continued compliance with these API Terms, we agree that you may use Uber Marks for certain limited purposes related to your Services as described below. These rights apply on a non-exclusive, non-transferable, worldwide, royalty-free basis, without any right to sub-license, and may be revoked by Uber at any time. If Uber updates any Uber Marks that you are using, you agree to update such Uber Marks to reflect the most current versions. You must not use any Uber Marks, or any confusingly similar mark, as the name or part of the name of your Services, or as part of any logo or branding for your Services.

本API使用协议的授权不包括与您服务相关的对Uber名称和任何Uber商标、服务商标或标志（下文简称“Uber商标”）的一般授权使用。如若您持续遵守本API使用协议，我们同意您基于特定有限目的在与您服务相关的下述情况下可以使用Uber商标。这些授权建立在非独占的、不可转让的、世界范围的、无版税的基础上，您没有任何再许可的权利，并且Uber可能随时撤销这些授权。如果Uber更新了您正在使用的任何Uber商标，您同意更新该Uber商标至现行最新版本。您不能使用Uber商标或任何混淆性相似标志作为您的服务名称或服务名称、服务标志、服务品牌的一部分。

Branding and Logos. We may occasionally provide branding elements and logos featuring certain Uber Marks for use within your Services, as further described here. You can use these resources as we provide, but you must not make any changes or modifications.

品牌与标志。我们有时可能提供构成特定Uber商标的品牌元素和标志让您在您的服务中使用，更详细的描述请点击这里。您可以使用我们提供的这些资源，但是您不可以做任何变动或修改。

Promoting Your App. You may not use any Uber Marks in any advertising, promotional or marketing materials for your Services, provided that you may use the name "Uber" in a written description of your Service in order to describe the fact that your Service connects to the Uber Services using the Uber API. The first time that the name appears in any piece of text, it should be followed by a superscript "Iregl" indicating a registered trade mark (i.e. "UBER Iregl"). You must not use the name in any way that takes unfair advantage of our goodwill or reputation in the Uber Marks, and you must not make any suggestion, directly or indirectly, that your app is an official Uber app or is endorsed by Uber in any way.

您的应用程序推广。您不能为了您的服务在任何广告、宣传、营销材料中使用Uber商标，除非您可能在书面描述您的服务时用到“Uber"的名字以陈述您的服务通过使用Uber API和Uber的服务有关联。"Uber"的名字在任何文本中第一次出现时，必须加上上角标"Iregl"表示这是注册商标（举例：”Uber Iregl")。您不能用任何方式使用Uber的名字以便从我们在Uber商标中的声誉或商誉中谋取不正当利益，并且您不能直接或间接地表示您的应用程序是Uber的官方应用程序或Uber以任何方式认可的应用程序。

Press and Publicity. You may not issue any press release or other announcement regarding your Services that makes any reference to Uber (except as permitted in the "Promoting Your App" paragraph above) without our prior written consent.

媒体与宣传。没有我们的事先书面同意，您可能不能进行任何提及您的服务和Uber有任何关联（在上述“您的应用程序推广”段落中允许的除外）的媒体发布或其他公告。

B. Ownership

B.　所有权

You agree that Uber and its affiliates retain all worldwide right, title and interest in and to the Uber API Materials, Uber Marks and the Uber Services, including, without limitation, all intellectual property rights therein. Any rights not expressly granted herein are withheld.

您同意Uber和它的关联企业保留所有的世界范围内的关于Uber API材料、Uber商标、Uber服务，包括但不限于，所有其中涉及的知识产权的权利、所有权和权益。任何本协议中未明确授权的权利亦被得到保留。

C. Independent Development
C.　独立开发

You understand that Uber may currently or in the future develop products and services that may be similar to or compete with your Services. Nothing in these API Terms shall in any way restrict Uber from pursuing any business activities or from entering into any agreement with any other person or company.

您了解Uber可能会于现在或将来开发和您的服务类似或有竞争性的产品和服务。本API使用协议的内容没有用任何方式限制Uber从事任何商业活动或限制Uber和任何个人或公司签订任何协议。

D. Feedback
D.　反馈

In the event that you choose to provide Uber with feedback, suggestions or comments regarding the Uber API Materials or your use thereof, you agree that Uber will be free to use, copy, modify, create derivative works, distribute, publicly display, publicly perform, grant sublicenses to, and otherwise exploit in any manner such feedback, suggestions or comments, for any and all purposes, with no obligation of any kind to you.

在您选择给Uber提供关于Uber API材料或者您对它们的使用的反馈、建议或意见的情况下，您同意Uber将会基于任何及一切目的免费使用、复制、修改、创作衍生作品、散播、公开展示、公开表演、再许可或者用其他任何方式利用这些反馈、建议或意见，并且对您不负有任何义务。

4. LEGAL TERMS
4. 法律条款

A. Confidentiality

You may be given access to certain information and software relating to the Uber API that is not generally known by the public ("Confidential Information"), which is confidential and proprietary to Uber. You agree to use the Confidential Information only for the purpose of using the Uber API in accordance with these API Terms, and you agree to not disclose any of the Confidential Information to any third party without Uber's prior written consent. You agree to protect the Confidential Information in the same manner that you would protect your own confidential and proprietary information but in no event using less than a reasonable degree of care.

A.保密

您可能被允许访问一些与Uber API有关的且通常不为公众所知的信息和软件（“保密信息”），这些信息和软件是保密的并且为Uber所有。您同意按照本API使用协议，仅仅以Uber API的使用为目的使用这些“保密信息”；您同意未经Uber事先书面同意，不透露任何“机密信息”给任何第三方。您同意您会以保护自己的私密和专有信息的方式保护前述“保密信息”，但在任何情况下都不能不尽到合理注意义务。

B. Termination

You may terminate these API Terms at any time by ceasing all use of the Uber API Materials and deleting all copies of the Uber API Materials (including, without limitation, all Uber Data) in your possession or control. We may immediately revoke your use of the Uber API without notice if that use violates any of these API Terms or if we otherwise object to your use of the Uber API. We may also terminate any rights granted hereunder, or may add or remove functionalities or features of the Uber API, at any time and for any reason at our discretion. We will not be liable to you or any third party for any costs or damages as a result of termination of this agreement.

B.协议终止

您可以通过停止使用您拥有或掌控的所有Uber API材料以及删除所有Uber API材料副本（包括但不限于所有的Uber数据）来随时终止本API使用协议。如果您对Uber API的使用违反本API使用协议的任何内容，我们可能随时取消您对Uber API的使用资格且不经另行通知，或者以其他方式反对您使用Uber API。

我们也可能出于我们的判断在任何时间以任何理由终结本协议项下授予您的任何权利，也可以增加或删除Uber API的功能或特征。对于终止本协议给您或任何第三方造成的任何费用或损失，我们将不负任何责任。

C. Disclaimer

THE UBER API, UBER API MATERIALS, AND ALL OTHER UBER SERVICES ARE PROVIDED "AS IS" AND ON AN "AS-AVAILABLE" BASIS, WITHOUT WARRANTY OF ANY KIND, AND UBER DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTIES OR CONDITIONS ARISING OUT OF COURSE OF DEALING OR USAGE OF TRADE. UBER DOES NOT WARRANT THAT THE UBER API, UBER API MATERIALS, OR ANY OTHER UBER SERVICES PROVIDED HEREUNDER WILL MEET YOUR REQUIREMENTS OR THAT USE THEREOF WILL BE ERROR-FREE, UNINTERRUPTED, VIRUS-FREE, OR SECURE.

C.免责声明

Uber API 、Uber API 材料，以及其它所有的Uber服务都是按照“现状”和“现有”的基础提供，不提供任何形式的担保；Uber API否认任何的担保，不论是明示的、暗示的还是法定的，更准确地说，包括但不限于适合销售性、不侵权、适用于某一特定用途的保证，以及其他任何产生于交易或商业惯例的保证或条件。Uber不保证Uber API 、Uber API 材料，以及其它所有本协议项下Uber提供的服务会符合您的要求，也不保证您对Uber API 、Uber API 材料，以及其它所有本协议项下Uber提供的服务的使用会是没有错误的，不间断的，无病毒的或绝对安全的。

D. Limitation of Liability

IN NO EVENT WILL UBER OR ITS EMPLOYEES, AGENTS, USERS OR PARTNERS BE LIABLE TO YOU FOR ANY SPECIAL, INCIDENTAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, LOSS OF USE, DATA, BUSINESS OR PROFITS) ARISING OUT OF OR IN CONNECTION WITH THESE API TERMS OR YOUR USE OF THE UBER API, UBER API MATERIALS, OR OTHER UBER

SERVICES, WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON CONTRACT, WARRANTY, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, AND WHETHER OR NOT YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. THE FOREGOING LIMITATIONS WILL SURVIVE AND APPLY EVEN IF ANY LIMITED REMEDY SPECIFIED IN THIS AGREEMENT IS FOUND TO HAVE FAILED ITS ESSENTIAL PURPOSE. UBER'S AGGREGATE LIABILITY UNDER THESE API TERMS WILL NOT EXCEED THE GREATER OF (i) THE FEES YOU PAID TO UBER FOR USE OF THE UBER API MATERIALS IN THE LAST YEAR OR (ii) ONE HUNDRED DOLLARS ($100).

在任何情况下，Uber或者其员工、代理商、用户或合作伙伴均不对任何特殊性的、偶然性的、惩罚性的或者间接性的损害赔偿（包括但不限于使用、数据、业务或利润损失）承担责任，不论这个损失是产生于或者与本 API使用协议相关，亦或产生于或者与您对Uber API 、Uber API 材料，以及其它Uber服务的使用相关，也不论这个责任是产生于任何基于合同、担保、侵权（包括但不限于疏忽）、严格责任等的索赔，更不论您是否已经被告知这种损失或损害的可能性。上述责任限制有效并适用于即使在本协议中明确的任何有限的补救被发现不能达到其根本目的的情况。Uber在本协议项下的责任总计不超过(i)您在过去一年中使用Uber API材料所支付的费用，或者(ii)一百美元（$100）。

E. Indemnification

You will indemnify, defend, and hold harmless Uber and its subsidiaries, affiliates, officers, directors, employees, agents, users and partners from any and all claims, damages, losses, liabilities, actions, judgments, costs, and expenses (including, without limitation, reasonable attorneys' fees) brought by a third party arising out of or in connection with: (a) your use of the Uber API, Uber API Materials or the Uber Marks other than as expressly allowed by this agreement; (b) your breach or alleged breach of any of the terms, conditions and representations under these API Terms; (c) your Services or business; and (d) your gross negligence or willful misconduct. You will control the defense and settlement of any claim subject to indemnification by you hereunder, provided that Uber may at any time elect to take over control of the defense and settlement of any claim. You may not settle any such claim without Uber's prior written consent.

E.赔偿

您同意确保Uber及其子公司，关联公司，管理人员，董事，雇员，代理，用户和合作伙伴免于承担因下列事项引起或与下列事项相关的第三方索赔、损害、损失、责任、诉讼、判决、诉讼费用、开支（包括但不限于合理的律师费用），对其进行赔偿，为其抗辩并使其免受损害：(a) 您用非本协议明示许可的方式使用Uber API, Uber API 材料或Uber marks； (b)您破坏或涉嫌破坏本API使用协议的任何条款、条件或陈述；(c) 您的服务或业务；(d) 您的重大疏忽或故意不当行为。您对本协议项下所有由您补偿的索赔的抗辩和和解享有控制权，前提是Uber可能随时选择接手控制这些索赔的抗辩或和解。未经Uber事先书面同意，您可能无权对此类索赔案件进行和解。

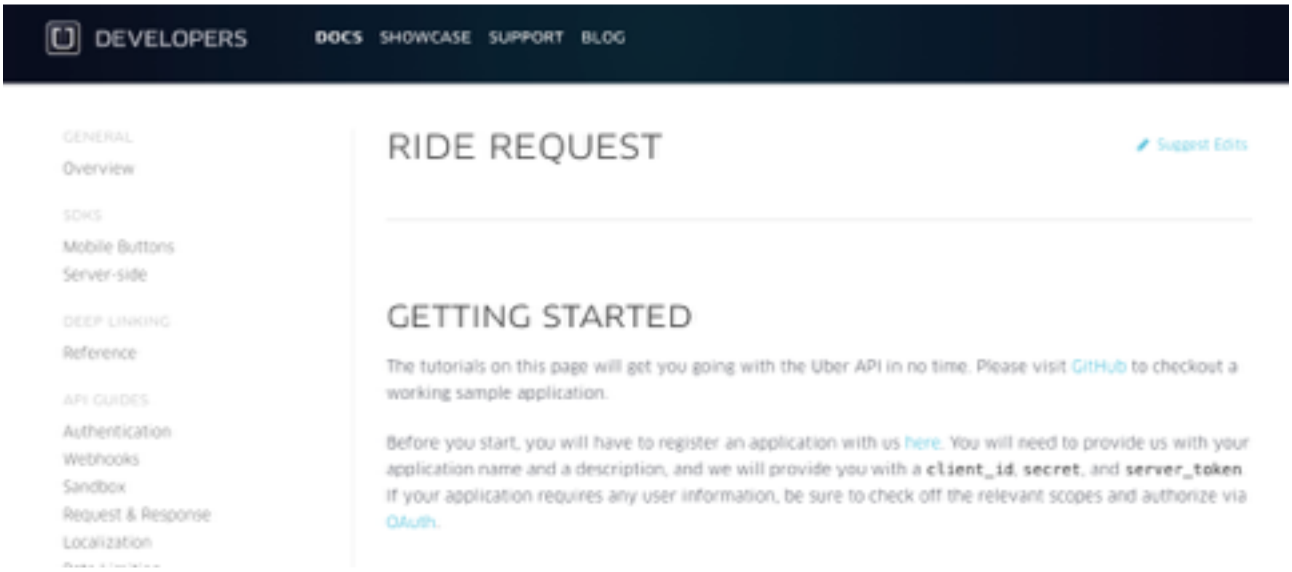
F. Miscellaneous

These API Terms will be governed by the laws of the State of California, without regard to conflicts of law principles. These API Terms were drafted in English and the English-language version shall control in the event of a conflict with any translated version. All claims arising out of or relating to these API Terms will be brought exclusively in the federal or state courts of San Francisco County, California, USA, and you consent to jurisdiction in those courts. You may not assign any of the rights or obligations under these API Terms, by operation of law or otherwise, without the prior written consent of Uber. Any attempted assignment in violation of this paragraph is void. These API Terms constitute the entire agreement among the parties with respect to the subject matter and supersedes and merges all prior proposals, understandings and contemporaneous communications. Any modification to the API Terms by you must be in a writing signed by both you and Uber.These API Terms do not create or imply any partnership, agency or joint venture between the parties. No waiver by Uber of any right under this agreement will be effective unless set forth in a writing authorized by Uber. If any part of these API Terms are determined to be invalid or unenforceable by a court of competent jurisdiction, that provision will be enforced to the maximum extent permissible and the remaining provisions of this agreement will remain in full force and effect.

F.其他

本API使用协议受到美国加利福利亚州的法律管辖，无需考虑法律冲突原则。本API使用协议是用英文起草，当其他语言版本协议与英文版本协议发生冲突时，以英文版本协议为准。所有由本API使用协议引起或与本API使用协议相关的权利主张均由美国加利福利亚州旧金山县的联邦法院或州法院享有管辖权，

您对上述法院享有上述管辖权表示同意。根据法律或其他规定，不经Uber事先书面同意，您不得转让本API使用协议项下的所有权利或义务。任何违反这段协议的企图转让都是无效的。本API使用协议是缔约方之间关于本协议标的的完整协议，取代并融合任何先前的提议或协定以及同时期的通信。任何您对本API使用协议的修改都必须是书面的且经您和Uber双方签字。本API使用协议不会在缔约方之间建立或暗示任何合伙、代理、合资关系。除非经Uber授权书面声明，任何Uber对本协议项下权利的放弃都不生效。如果本API使用协议的任何部分经具有司法管辖权的法院认定无效或不可强制执行，则该等条款将会在最大的允许范围内执行，并且本协议的其他条款会继续具有完全效力。



RIDE REQUEST

GETTING STARTED

The tutorials on this page will get you going with the Uber API in no time. Please visit GitHub to checkout a working sample application.

Before you start, you will have to register an application with us here. You will need to provide us with your application name and a description, and we will provide you with a `client_id`, `secret`, and `server_token` If your application requires any user information, be sure to check off the relevant scopes and authorize via OAuth.

RIDE REQUEST
行程请求

GETTING STARTED
开始

The tutorials on this page will get you going with the Uber API in no time. Please visit GitHub to checkout a working sample application.
该页面上的教程会立刻带你了解Uber API的开发。请访问GitHub以获取可工作的应用程序示例。

Before you start, you will have to register an application with us here. You will need to provide us with your application name and a description, and we will provide you with a `client_id`, `secret`, and `server_token`. If your application requires any user information, be sure to check off the relevant scopes and authorize via OAuth.
在你开始之前，你必须在这里（here）向我们注册一个应用程序，你会需要提供给我们你的应用程序的名字和描述，同时我们会提供给你对应的client_id，secret和server_token。如果你的应用程序需要任何用户信息，请通过OAuth确认相关的域和授权。

SERVER SIDE AUTHENTICATION
服务侧鉴权

The server side authentication tutorial will go through using the Uber API endpoints for requests that do not require a user login. These endpoints include:
服务侧鉴权教程会向你介绍为无须用户登录即可发起请求的Uber API终端接口，这些终端接口包括：

```
GET /v1/products
GET /v1/estimates/time
GET /v1/estimates/price
```

MAKING A REQUEST
发起一个请求

This is an example request made with server_token authentication. In this example, we are making a call to the products endpoint which will return all of the Uber products that are available at the specified latitude and longitude.
这是一个使用server_token鉴权的请求的示例。在这个示例当中，我们正在向products终端接口发起一个请求，它会返回所有指定经纬度地点可用的Uber产品服务。

```
import requests
```

```
url = 'https://api.uber.com/v1/products'

parameters = {
    'server_token': 'INSERT_SERVER_TOKEN_HERE',
    'latitude': 37.775818,
    'longitude': -122.418028,
}

response = requests.get(url, params=parameters)

data = response.json()
```

OAUTH 2.0

The server OAuth tutorial will go through using the Uber API endpoints for requests that require permission of an authenticated user. The Uber API uses OAuth 2.0 and for this Python example we will use the third-party libraries OAuth2Service (from the rauth) and requests.
服务器OAuth教程会向你介绍需要一个鉴权用户的许可才能发起请求的Uber API终端接口。Uber API使用OAuth 2.0，且在这个Python示例当中我们会使用第三方库OAuth2Service（来自于rauth）和requests。

The endpoints that require OAuth include:
需要OAuth的终端接口包括：

```
GET /v1/me
GET /v1.1/history
GET /v1/requests
```

Step 1
第一步
Have the user sign in, authorizing your app to access their Uber account.
让用户登录，授权你的应用访问他们的Uber账户。

```
from rauth import OAuth2Service

 uber_api = OAuth2Service(
     client_id='INSERT_CLIENT_ID',
     client_secret='INSERT_CLIENT_SECRET',
     name='INSERT_APP_NAME',
     authorize_url='https://login.uber.com/oauth/authorize',
     access_token_url='https://login.uber.com/oauth/token',
     base_url='https://api.uber.com/v1/',
 )
```

```
parameters = {
    'response_type': 'code',
    'redirect_uri': 'INSERT_ROUTE_TO_STEP_TWO',
    'scope': 'profile',
}

# Redirect user here to authorize your application
login_url = uber_api.get_authorize_url(**parameters)
```

Step 2
第二步
After you have authorized, the authorization code will be exchanged for an access token. This access token will be required for calling the different parts of the API.
在你完成授权之后，授权码会被交换为一个access token，在调用不同部分的API时需要该access token。

```
import requests

# Once your user has signed in using the previous step you
should redirect
# them here

parameters = {
    'redirect_uri': 'INSERT_ROUTE_TO_STEP_TWO',
    'code': request.args.get('code'),
    'grant_type': 'authorization_code',
}

response = requests.post(
    'https://login.uber.com/oauth/token',
    auth=(
        'INSERT_CLIENT_ID',
        'INSERT_CLIENT_SECRET',
    ),
    data=parameters,
)

# This access_token is what we'll use to make requests in the
following
# steps
access_token = response.json().get('access_token')
```

The access token is what you will need to make requests on behalf of the logged in user. You should store this token securely, so that the user will not have to go through the authorization process again.

在你代表一个已登录用户发起请求时，需要此access token，你需要安全地存储这个token，从而使用户不需要重复进行授权流程。

Step 3
第三步
Now that we have an access token, we will be able to make requests to the Uber API. The following is an example call to the `/v1/me` endpoint that will return user information about the authorized user.
现在我们拥有了access token，我们可以向Uber API发起请求了。下面是一个调用 /v1/me 终端接口的示例，它将会返回授权用户相关的用户信息。

Request
请求

```
url = 'https://api.uber.com/v1/me'
response = requests.get(
    url,
    headers={
        'Authorization': 'Bearer %s' % access_token
    }
)
data = response.json()
```

Here is an example of the data that would have been returned form the previous step:
这是进行完上一步之后有可能得到的返回数据的示例：

Response
响应

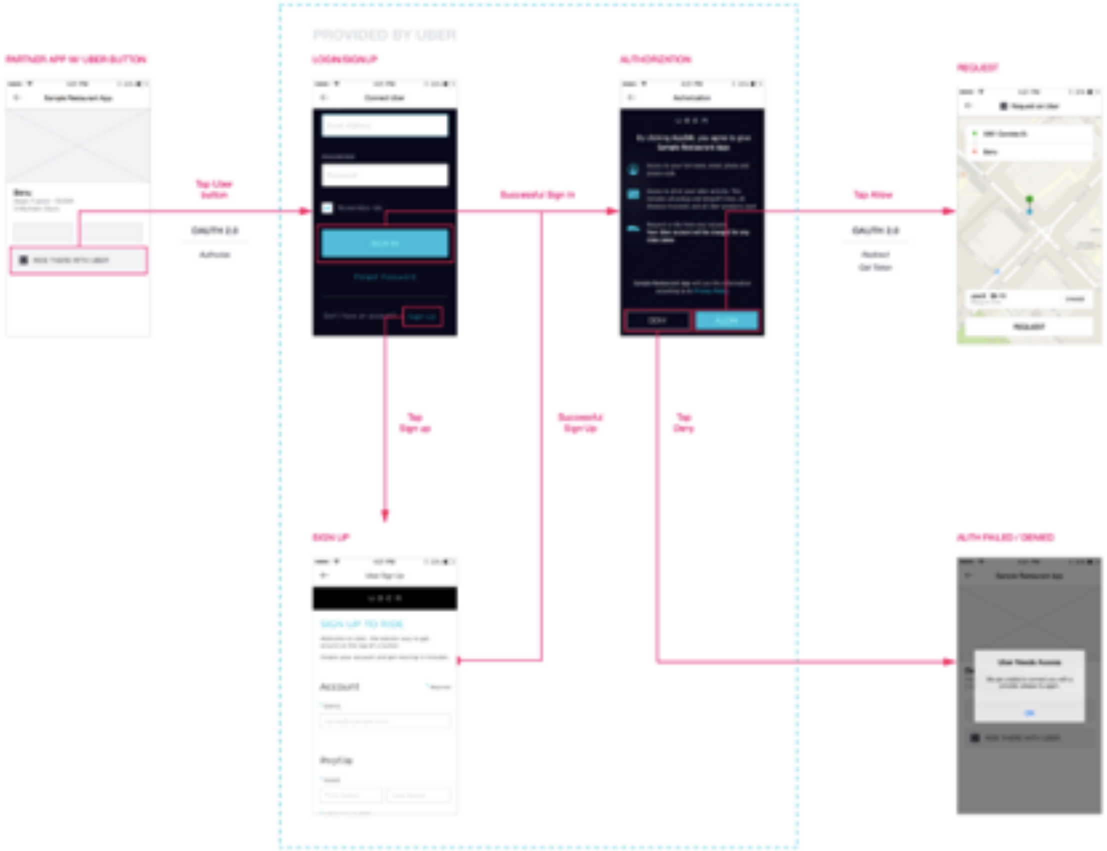
```
{
    "first_name": "Uber",
    "last_name": "Developer",
    "email": "developer@uber.com",
    "picture": "https://...",
    "promo_code": "teypo"
}
```

Authentication and Sign Up Flow
鉴权与注册流程

REQUEST

请求

The Uber API gives you the ability to Request an Uber Product on behalf of users within your application. Given you know where a user currently is,



where they want to go, and which Uber product they want to use to get there, you have all of the tools to make that happen with a few simple API endpoints.
Uber API给予你在你的应用内代表用户发起一个Uber产品服务的行程请求（Request）的能力。通过给出你所知的用户当前所在的位置，用户想要去的地点，以及用户想要使用的Uber产品服务，你就拥有了通过一些简单的API终端接口实现行程的所有工具。

Note: These endpoints will make requests to active drivers on the Uber platform, directing them to drive to the locations of users who will be charged for all said activity.
注意：这些终端接口会向Uber平台的活跃司机发起请求，引导他们驾车前往用户所在的地点，用户将会因为所有发起的行为被收费。

While we are giving developers a lot of freedom on how to build interfaces and present information to the users of their applications, there are a few essentials that must be included to ensure a user has all necessary information. This tutorial aims to walk through all of the required Uber API Request calls, how they interact with one another, and provide best practices for building an intuitive user experience.

在我们给予开发者在他们自己的应用内部极大程度自由来建立交互界面和展示信息的时候，有一些必须的重要信息是必须被包括在内的，以保证用户知晓全部必要的信息。该教程会引导你学习所有需要的Uber API行程请求接口调用，以及他们如何与其它的接口进行交互的，同时，还会提供构建最好用户体验的最佳实践。

THE SANDBOX
沙盒

Because of the real-world nature of the Request endpoints, which call active drivers on the system which result in financial transactions on Uber rider accounts, the Uber API provides a Sandbox environment for testing. All Requests made to the API Sandbox will result in a simulated Request that can be programmatically updated.

因为线上行程请求接口的特性，将会呼叫系统中的活跃司机，并且会对Uber乘客的账户产生交易，Uber API提供了一个沙盒测试环境。所有发往沙盒API的行程请求都会形成一个仿真的，可以通过编程更新的行程请求。

To get an understanding on how to set up your development environment setup for making Requests within the sandbox environment, see the Sandbox Documentation.

要理解如何设置你的开发环境来对沙盒环境发起行程请求，请查阅沙盒部分文档（Sandbox Documentation）。

THE BASICS
基础

There are essentially four things the Request endpoints enable your application to do:

通过行程请求终端接口，你可以完成四项重要的基础事项：

- Make a Request - POSTing a start lat/lng, end lat/lng, and a `product_id` gets a Request kicked off
- Make a Request（发起行程请求）- POST一个起点经纬度和一个终点经纬度，一个product_id来开始一个行程请求。
- Poll Request Status - once a Request has been created, you can GET that status of that Request, accompanied by other Request details, to keep your user in the loop

- Poll Request Status（轮询行程请求状态）- 当一个行程请求被创建，你可以GET到该行程请求当前的状态，以及其它相关细节信息，来保持你的用户在进程当中。
- Cancel a Request - Sometimes a user needs to cancel a request and this is done by simpling issuing a DELETE command
- Cancel a Request（取消行程请求）- 有些时候用户需要取消行程请求，此时，只要简单地发送一个DELETE（删除）命令即可。
- Provide a Map - If you would like to show a visual representation of the Request
- Provide a Map（提供地图）- 当你想要向用户展示一个行程请求对应的视觉展示。

LIFE CYCLE OF A REQUEST 一个行程请求的生命周期

A Request can go through many states between its start and end. We indicate this state as the status attribute returned by the Request endpoint. The possible values for status are:
一个行程请求在从开始到结束的过程里会经历很多的状态。我们通过行程请求接口返回的状态参数来表示当前所处的状态。可能的状态的值有：

状态	描述
`processing`	The Request is matching to the most efficient available driver. 该行程请求正在匹配最合适的空闲司机。
`no_drivers_available`	The Request was unfulfilled because no drivers were available. 该行程请求未被满足，因为没有空闲司机可以指派。
`accepted`	The Request has been accepted by a driver and is "en route" to the start location (i.e. `start_latitude` and `start_longitude`). 该行程请求被某位司机接受，且司机“正在途中”前往起始地点（即start_latitude和start_longitude）。
`arriving`	The driver has arrived or will be shortly. 司机已经或者即将抵达。

状态	描述
`in_progress`	The Request is "en route" from the start location to the end location. 该行程请求“正在途中”，从起始点前往终点。
`driver_canceled`	The Request has been canceled by the driver. 该行程请求被司机取消。
`rider_canceled`	The Request canceled by rider. 该行程请求被乘客取消。
`completed`	Request has been completed by the driver. 该行程请求已经被司机完成。

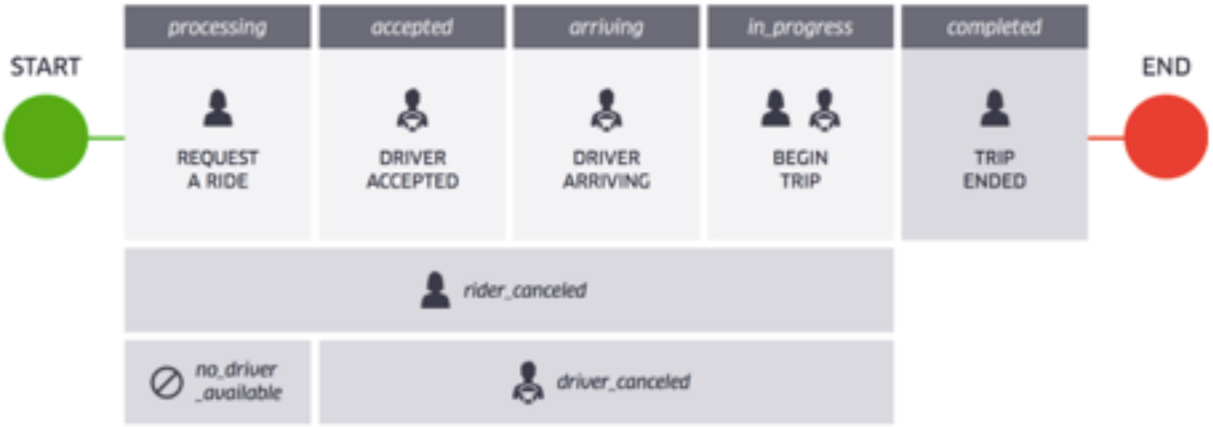
Overall Request Flow
总行程请求流程

AUTHORIZATION AND AUTHENTICATION
授权与鉴权

Applications that wish to take advantage of the Request endpoints must have their users authorize their application to make requests on their behalf. This is done by having a user grant access of the request scope to your application via OAuth2.0. See the OAuth 2.0 section of our tutorials for more details.
希望使用行程请求终端接口的应用必须让他们的用户来授权他们代表其发起请求。这个过程是通过OAuth 2.0来让用户完成授权你的应用程序访问特定所需要访问的域来实现的。请参看我们的教程的OAuth 2.0部分来获得更多细节。

Once a user has effectively authorized your application, all requests to the Reqeust endpoint must be made with an OAuth bearer token.
当一个用户有效授权了你的应用时，所有向行程请求终端接口发起的请求都必须使用OAuth 2.0不记名token。

Applications using the `request` scope must be approved by Uber before they can be made available to the general public. By default, only the developer and invited application admins may make Requests.



使用request域的应用必须在公开向大众提供服务之前，得到Uber的许可（https://developer.uber.com/support#access）。默认情况下，只有开发者和受邀请的应用程序管理员可以发起行程请求。

PREPARING TO MAKE A REQUEST 准备发起行程请求

Once a user has granted your application permission to make Requests on their behalf, you only need three pieces of information to do so:
当一个用户授权你的应用程序代表其发起行程请求，你只需要以下三个部分的信息来完成请求：

- Product - the product_id of the Product the user wishes to have requested.
- 产品服务 - 用户想要请求的产品服务对应的product_id。
- Start Location - the latitude and longitude pair of where the user would like to be picked up (usually their current location).
- 起点位置 - 用户想要被接载的地点的经纬度（通常是他们的当前位置）
- End Location - the latitude and longitude pair of where the user would like to be dropped off.
- 终点位置 - 用户想要到达的终点的经纬度。

There are typically three items a user wants to know about a Product before choosing one for making a Request:
有三个典型信息是用户在选择一个产品服务发起行程请求前想要知道的：

- Products Available - given a location, the Uber Products a user can take advantage of, provided by the Products endpoint.

- 可用的产品服务 - 给定地理位置，用户可以享用的Uber产品服务，由Products终端接口提供。
- Estimated Time of Arrival - how quickly a particular product get to your user, provided by the Time Estimates endpoint.
- 预计到达时间 - 特定产品服务多快能够接到你的用户，由 Time Estimates 终端接口提供。
- Estimated Cost of Service - how much a user will be charged to go from the start to the end of a Request, provided by the Price Estimates endpoint.
- 预计服务费用 - 一个行程请求从起点到终点需要花费用户多少费用，由Price Estimates终端接口提供。

Once you've collected the above info, you can present these details to your user to help them select which product is right for them.
当你收集到了上述信息，你就可以给你的用户展示这些细节，以帮助他们选择合适的产品服务

To get the user's start and end locations, there are quite a few different approaches. Depending on the platform you are developing for, you may get the current location of a user by asking their device at the system level. Perhaps you have a list of recommend locations they can chose. You could also take the same approach as the Uber app and give them a pin on a map that can be dragged and dropped to specify location.
要获得用户的起点和终点位置，有一些不同的获取方式。取决于你正在开发的平台，你可以通过在系统层面向用户的设备询问当前的位置信息，你或许可以给用户提供一系列可供选择的推荐地点。你也可以使用和Uber应用一样的方式，给予用户一个在地图上定位的大头针，让用户拖动并且放置在特定的起点位置上。

CREATING A REQUEST
创建行程请求

Now that you have the product they wish to use and the start and end locations of the Request they would like to make, we have everything we need. Take those values and make a POST on behalf of the user to the Request endpoint.
现在，你有了用户想要使用的Uber产品服务类型，和用户想要发起的行程请求对应的起点位置和终点位置，我们就有了所需要的一切。使用这些值，代表用户向Request终端接口发起一个POST请求。

```
curl -v -H "Authorization: Bearer <OAUTH TOKEN>" \
     -H "Content-Type: application/json" -X POST -d \

'{"start_latitude":"37.334381","start_longitude":"-121.89432"
,"end_latitude":"37.77703","end_longitude":"-122.419571","pro
duct_id":"a1111c8c-c720-46c3-8534-2fcdd730040d"}' \
      https://sandbox-api.uber.com/v1/requests
```

This POST will result in three possible responses:
该POST请求有可能导致以下三种回应：

- 202 Accepted - Your Request is successfully being processed.
- 202 接受 - 你的行程请求已经递交成功，正在处理中。
- 409 Conflict - An error has occurred, possibly due to Surge Pricing or no drivers available.
- 409 冲突 - 发生了一个错误，可能是因为提价（Surge Pricing）或者是没有空闲司机。
- 422 Unprocessable Entity - An error has occurred, most likely due to an issue with the user’s Uber account.
- 422 无法处理 - 发生了一个错误，很可能是因为用户的Uber账户存在异常。

In most cases, as long as you provide a valid product, start location, and end location, you will get a 202 Accepted response letting you know that the Request was created successfully.
在大多数情况下，只要你提供了有效的产品服务类型、起点位置和终点位置，你就会得到“202 接受”响应，让你知道该行程请求已被成功创建。

```
Status-code 202 Accepted
{
  "request_id":"852b8fdd-4369-4659-9628-e122662ad257",
  "status":"processing",
  "vehicle": null,
  "driver": null,
  "location": null,
  "eta": 5,
}
```

In the cases where you receive a `409 Conflict` for surge pricing, please see our Handling Surge Pricing section of the tutorial.
在你收到因提价产生的“409 冲突”的情况下，请参考我们的处理提价（Handling Surge Pricing）部分教程文档。

PROCESSING A REQUEST

处理行程请求

Now that you have successfully created a Request and have a `request_id` that was returned in the response to your POST, you can continuously poll the Uber API to get the current status and additional details of a Request. We recommend you do this every 3-5 seconds to give the most up-to-date details possible. See Displaying Details of Request for more details.

现在你已经成功创建了一个行程请求，也拥有了一个对应的在你发起了POST请求之后在响应中返回的request_id，你可以继续轮询Uber API，来获取当前的状态和一些行程请求的额外细节。我们建议你没3-5秒轮询一次，以获得最有时效性的细节。请查阅行程请求的展示细节信息部分来获得更多细节。

```
curl -v -X "GET" "https://sandbox-api.uber.com/v1/requests/<REQUEST_ID>" \
  -H "Authorization: Bearer <OAUTH TOKEN>"
```

Details you will be provided include the the driver’s name, picture, and phone number as well as the location, make, model and license plate of the vehicle. The more of these details you provide, the better the experience the user will have and the more likely they will successfully complete a Request.

你需要展示的细节包括司机的名字、照片，以及电话号码、定位地址、车辆的制造厂家、型号、车牌号。你提供的细节越多，用户就会得到更佳的使用体验，这样他们也更容易成功完成一次行程请求。

```
Status-Code: 200 OK
{
   "status": "accepted",
   "driver": {
      "phone_number": "(555)555-5555",
      "rating": 5,
      "picture_url": "https:\/\/d1w2poirtb3as9.cloudfront.net\/img.jpeg",
      "name": "Bob"
   },
   "eta": 4,
   "location": {
      "latitude": 37.776033,
      "longitude": -122.418143,
      "bearing": 33
   },
   "vehicle": {
      "make": "Bugatti",
      "model": "Veyron",
      "license_plate": "I<3Uber",
      "picture_url": "https:\/\/d1w2poirtb3as9.cloudfront.net\/car.jpeg",
   },
   "surge_multiplier":  1.0,
   "request_id": "b2205127-a334-4df4-b1ba-fc9f28f56c96"
}
```

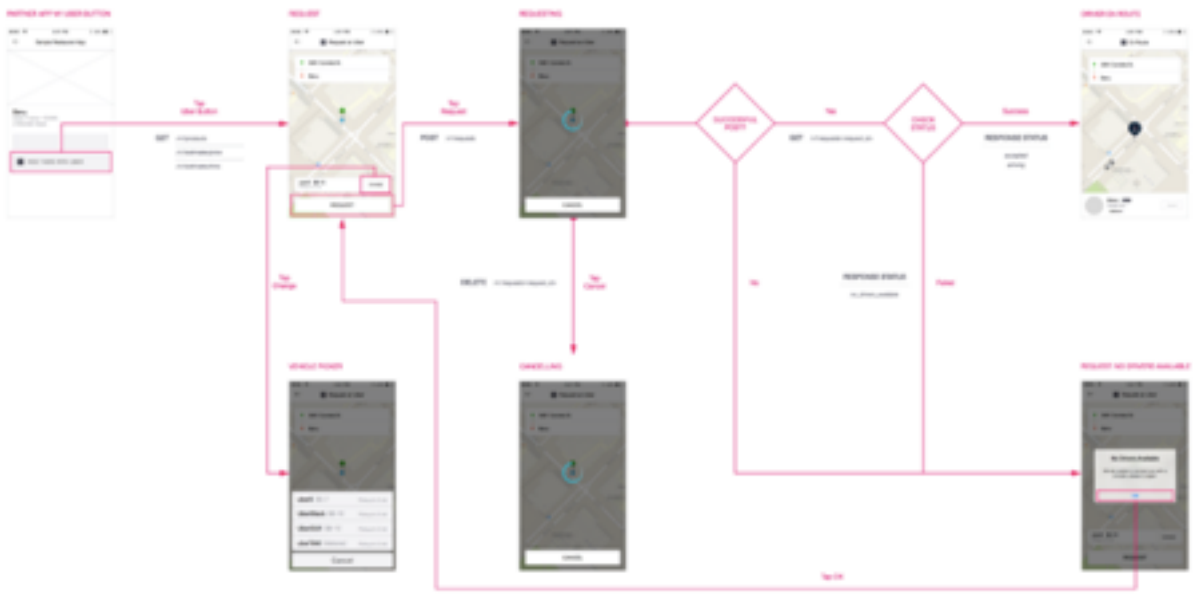
CANCELLING A REQUEST
取消行程请求

If a user decides to cancel a Request, you must provide the ability to cancel that Request within your application.
如果用户决定取消行程请求，你必须在你的应用里提供取消该行程请求的能力。

To do this, simply send a [DELETE to the Request endpoint] with the specified `request_id` and ensure a successful response is returned. This is usually the result of a user pressing a ‘Cancel’ button and confirming within a modal.
要做到这一点，简单地向行程请求接口发送一个DELETE，带上相应的request_id参数，保证得到返回的成功响应即可。这个过程通常是由用户点击一个“取消”按钮后接着在一个模态页面确认。

```
curl -X "DELETE" "https://sandbox-api.uber.com/v1/requests/
<REQUEST_ID>" \
  -H "Authorization: Bearer <OAUTH TOKEN>"
```

Request Initiation and Processing
行程请求初始化和处理



DISPLAYING DETAILS OF REQUEST 展示行程请求详情

As a Request is ongoing, you will want to display the details of the Request to the user in a way that helps them understand what is happening and give them the information they need to find or get in touch with their driver.
当一个行程请求正在进行时，你会想要向用户展示该行程的详情，以帮助他们了解现在的状况，给他们展示相应的信息，让他们能找到或者联系上他们的司机。

For instance, when a Request is `processing` it's probably best to let the user know Uber is attempting to find them a driver. Using a spinner or other loading indicator can convey this message well.
例如，当一个行程请求正在processing状态时，最好让用户知道Uber正在尝试为他们寻找到一个司机。使用一个旋转指示器（spinner）或者是其它类型的加载指示器就可以很好的传达这个信息。

Once a Request status has changed to `accepted` you can let your user know a vehicle is enroute, what the details of the driver are, and remind them their pickup location. You can even show them the location of the vehicle on a map because the Uber API provides this location information.
一旦一个行程请求的状态被切换到accepted，你就能让你的用户知道车辆已经在接载的路上，还能让用户知道司机相关的详情，同时提醒他们接载地点的位置。你甚至能在地图上给他们展示车辆的位置，因为Uber API会提供相应的位置信息。

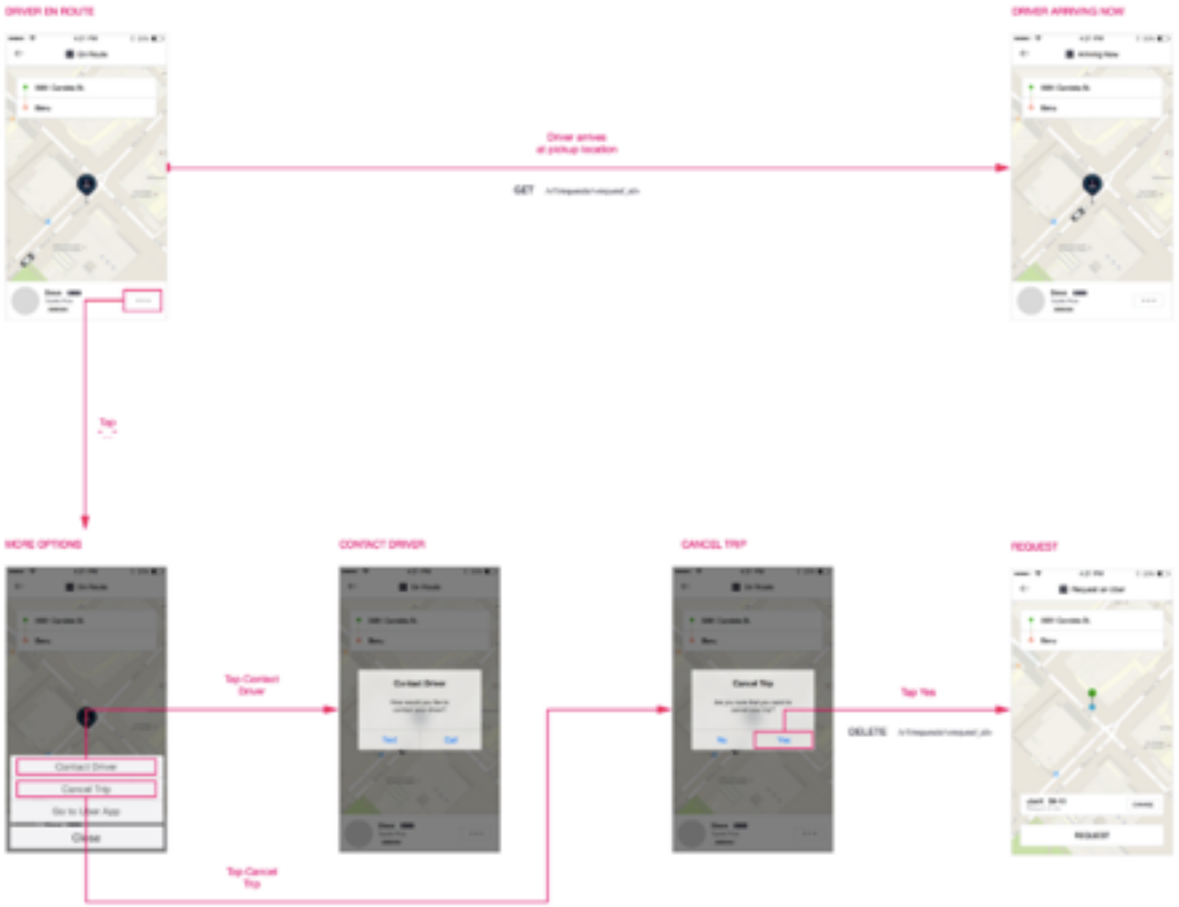
If the Request status becomes `arriving` you probably want to indicate this to the user in a way that gets their attention. And once a Request is `in_progress` you can provide the information a user might find useful once they are already in the vehicle.
如果行程的状态变为arriving，你可能会想要通过某种方式告知用户，引起他们的注意。而如果一个行程的状态变为in_progress，则你可以提供一些用户在车里时可能会觉得有用的信息

Unfortunately, there are some times when a driver must cancel a Request, and your application should let the user know when this happens. This will be indicated when the status of a Request is returned as `driver_canceled.` At this point it's usually helpful to make it easy for a user to re attempt a Request.

不幸的是，有些时候司机不得不取消行程，你的应用应当让用户知道发生了这样的情况。你可以通过行程的状态变为driver_canceled来了解到这种情况的出现，此时应当让用户很容易地再次发起一次行程请求。

Lastly, sometimes a user needs to get in touch with a driver to clarify their location, so please provide an easy way to call or SMS the driver from within your application.
最后，有些时候用户需要和司机取得联系，以说明他们所处的地理位置，所以请提供一个方便的路径让用户可以从你的应用中呼叫或者发送短信给司机。

Request Accepted Flow
行程请求被接受流程



INDICATING RIDE STATUS THROUGHOUT APPLICATION

通过应用程序展示行程状态

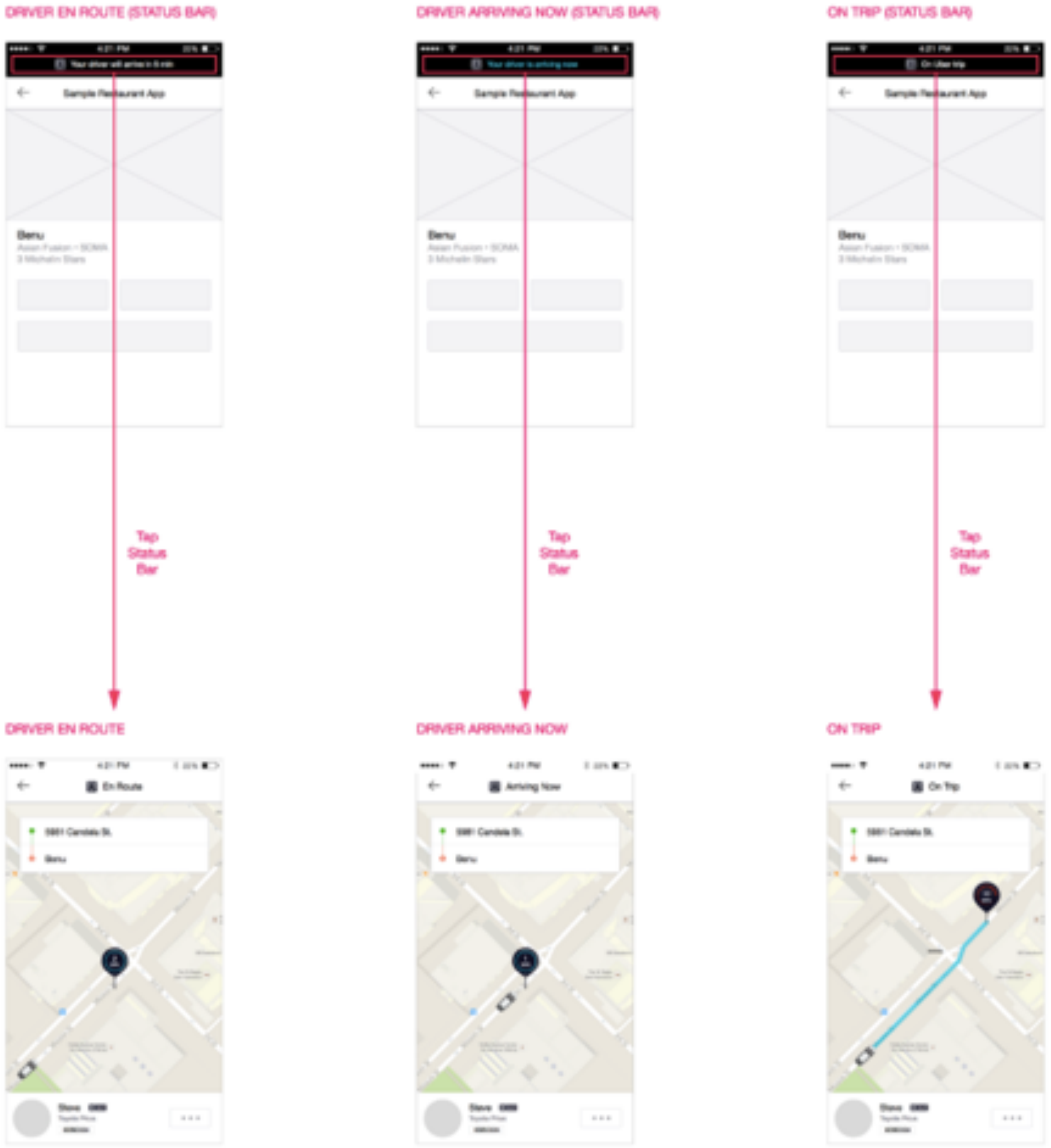
One of the benefits of integrating the Request endpoints within your application is that while a Request is ongoing, you can continue to interact with your user without them leaving your app.
在你的应用里集成行程请求接口的一个好处就是，当行程正在进行时，你可以持续地和你的用户交互，而无需让他们离开你的应用。

To take advantage of this, we recommend you present the status of an ongoing Request throughout your application. This provides Request status at a quick glance as well as easy access to more details and Request actions.
想要利用这个优势，我们推荐你通过你的应用程序持续展示正在进行的行程的状态。你可以提供一个简单的一眼即可了解的行程状态，以及能够轻松进入更多详情和行程选项的页面的路径。

The most common statuses and information users are interested are:
通常，用户最感兴趣的状态和信息是：

- Accepted - The driver is en route, so its a great place to show an ETA
- 已接受 - 司机正在前来接载的途中，所以这是一个展示估计到达时间（ETA）的最佳位置。
- Arriving - The driver is arriving now! Let your users know its time to find their driver
- 即将抵达 - 司机即将抵达！让你的用户知道是时候寻找他们的司机了。
- On Trip - The user is in a vehicle and off to their destination.
- 行程中 - 用户已经上车并且前往目的地。

Indicate Ride Status
展示行程状态



DRIVER EN ROUTE (STATUS BAR)

DRIVER ARRIVING NOW (STATUS BAR)

ON TRIP (STATUS BAR)

Tap Status Bar

Tap Status Bar

Tap Status Bar

DRIVER EN ROUTE

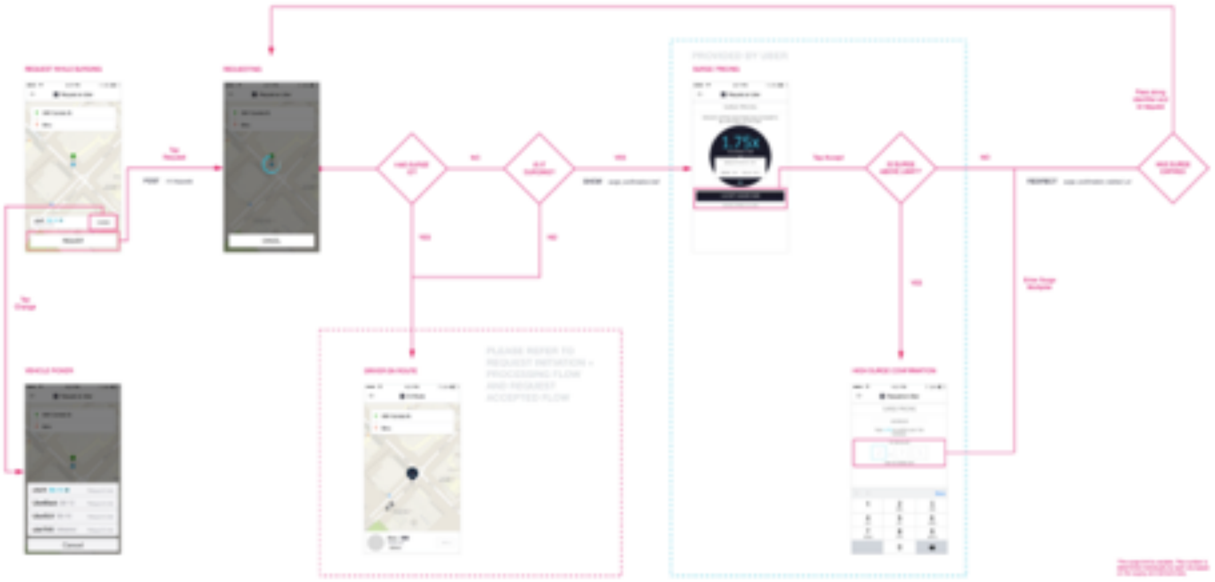
DRIVER ARRIVING NOW

ON TRIP

HANDLING SURGE PRICING
处理提价

The way a user accepts surge pricing is similar to how they authorize your application to use the request scope. We provide you with an HREF you send the user to, they accept the the surge multiplier by submitting a web form, and then we redirect back to your application with a surge_confirmation_id that you use to successfully make a Request.
用户同意提价的方式和让他们授权你的应用使用行程请求域很类似，我们想你提供一个HREF，你将用户送到这个页面，然后他们通过提交一个网页表单同意提价因子，接下来我们会带着一个surge_confirmation_id重定向返回你的应用程序，你将使用该参数成功发起一个行程请求。

Surge Flow
提价流程


PROVIDED BY UBER
1.75x
PLEASE REFER TO REQUEST INITIATION + PROCESSING FLOW AND REQUEST ACCEPTED FLOW

While most of the time making a Request is as easy as providing product and location information, there is a chance that a product with have surge pricing in effect. With surge pricing, Uber rates increase to get more cars on the road and ensure reliability during the busiest times. When enough cars are on the road, prices go back down to normal levels.
绝大多数时候，发起一次行程请求都非常简单，只需提供产品服务类型和地理位置信息，但有些时候，一些产品服务会存在提价的情况。当出现提价时，

Uber的费率会增加，以便吸引更多的车辆上路加入服务，并且保证繁忙的时段的服务稳定性。当足够的车辆在路上时，价格会回落到正常水平。

To ensure your users are fully informed about the current pricing of a Product, the Uber API will not allow a Request to be made without the user accepting the current surge price multiple.
为了保证你的用户能完全理解现在的产品服务的价格，Uber API不会允许一个行程请求在没有得到用户同意当前提价因子的情况下发起。

To test this, you need to enable surge pricing by making a PUT to the Products Sandbox endpoint with the desired `surge_multiplier`. Anything above 1.0 will activate surge pricing for a particular Product, which turns on the surge confirmation flow when making a Request for that Product in the Sandbox.
如果需要测试，你需要发送一个PUT命令给Products沙盒接口，带上想要的surge_multiplier。任何大于1.0的数值都会激活某个特定产品服务的提价状态，之后，当向Product沙盒接口发起对应产品服务的行程请求时，会开启一次提价确认流程，

When surge pricing is in effect, a POST to create a Request will be returned with a `409 Conflict` error that contains a `surge_confirmation` object within the `meta` component of the response; there are two values present, which are `surge_confirmation_id` and `href.`
当提价生效时，创建行程请求的POST请求将返回409 Conflict错误，其中会包含一个surge_confirmation对象，对象内包含响应的meta构件；一共有两个值会被展示，分别为surge_confirmation_id和href。

For a user to accept surge pricing, you must send them to the HREF within a web browser. They will be presented with a surge confirmation screen where they must accept the surge multiplier by submitting a web form. In some cases, when the surge multiplier is above a certain limit (most of the time when 2.0x or above), the confirmation page will be a two step process.
为了让用户同意提价，你必须带他们前往一个通过网页浏览器打开的HREF页面。一个提价确认页面会被展示给他们，在页面上，用户必须通过提交一个网页表单的形式来同意提价因子。有些时候，当提价因子超过某个特定值时（通常是2.0x或者更高），确认页面会有两部验证的流程。

Once a user successfully accepts the surge pricing multiple, they will be redirected to your application via `SURGE_CONFIRMATION_REDIRECT_URI`

(that is configurable via the Application Dashboard) which will include the `surge_confirmation_id` as a querystring parameter.
当用户成功同意提价时，他们会通过SURGE_CONFIRMATION_REDIRECT_URI（这是可以通过你的应用程序面板修改的）被重定向到你的应用程序，同时还会包括surge_confirmation_id作为查询字符串参数。

Now that the user has accepted surge and has returned to your application, you may make a second POST to the Request endpoint with the `surge_confirmation_id` parameter filled in with the appropriate identifier.
现在，用户已经同意了提价，并且已经返回了你的应用程序，你可以像行程请求接口（Request）发布第二个POST请求，包括surge_confirmation_id参数内填写了合适的标识符。

**Note: Because surge pricing fluctuates over time, there is a chance that a surge_confirmation_id will expire. In this case, th
注意：因为提价一直在变动，所以有些时候surge_confirmation_id有可能过期失效。

ALL REQUESTS AVAILABLE IN OFFICIAL UBER APPS
所有的行程请求在官方Uber应用里都可以查看

Since a Request that is made in a 3rd party application is done on behalf of an Uber rider, the status and all details of those Requests will be available within the official Uber apps. This means that if a user chooses to open the Uber app after making a Request with your application, they will have complete functionality as if they made the Request normally.
鉴于行程请求是由第三方程序代表Uber乘客发起，这些行程的所有状态以及详情，都可以在Uber官方应用当中查看到。这也就意味着，如果用户在通过你的应用发起了行程请求后，选择打开Uber应用，他们会拥有和平常正常发起行程请求一样的全部功能。

This also means that a user can perform the standard actions against a Request within an official Uber app, including canceling. Because of this, your

application should always rely on the information provided by the Request endpoint to provide the real time status and details of a Request.
这同时也意味着一个用户可以在Uber应用中针对某个行程请求进行标准的操作，包括取消行程。正因为如此，你的应用应该时刻通过Request终端接口提供的信息来保持行程请求的状态和详情的同步。

NOTIFICATIONS
通知

All notifications that a user would receive by Uber, as if they requested using the official Uber app, will be sent for Requests initiated by 3rd party applications. This includes, but is not limited to:
所有的用户可能收到的来自Uber的通知，就像他们在Uber官方应用里发起行程请求的情况一样，通过第三方应用发起的行程也会触发这样的通知发送给用户。这些通知包括但不限于：

- Push or SMS notifications regarding a Request status
- 关于行程状态的推送通知或者短信通知
- Email receipts
- 通过Email寄送的收据

Because Uber is handling notifications to the user about Requests, we recommend only showing in-app notifications about Request status so that a user isn’t bombarded with duplicate messaging. See Indicating Ride Status Throughout Application for some example in-app status indicators.
因为Uber处理了给用户的行程相关的通知，所以我们推荐你只使用应用内通知，来标识行程状态，这样用户就不会重复的消息进行轰炸了。请查看“通过应用程序展示行程状态”来参考某些应用内状态指示器的示例。

LIMITATIONS
限制

While the current endpoints give you the essentials for requesting trips, there are a few interactions that cannot be completed by the Uber API. These include:
当现有的终端接口能够给你完成一次行程请求所必须的一切，仍然有一些交互是通过Uber API无法完成的，包括：

- Rating drivers - Requests made and completed cannot be given a rating by a user via the Uber API. All ratings must be done by the Uber app
- 评价司机 - 通过Uber API，用户无法给已完成的行程进行评分，所有的评分都必须在Uber应用中完成。
- Displaying receipts - The fare details of a Request are not provided by the Uber API but all Requests resulting in a charge to a user will produce a receipt emailed to the user's email address associated with their Uber account
- 展示收据 - 行程请求的费用详情无法通过Uber API提供，但是所有的行程最终产生的对用户的收费都会给用户Uber账户绑定的邮箱发送一封含有收据的邮件。
- Splitting the fare - There is no way to initiate a Split Fare request through the Uber API, but users can continue to utilize this feature within the Uber app during the lifecycle of a request
- 分摊费用 - 通过Uber API无法发起一个分摊费用的请求，但是当行程处于特定生命周期时，用户可以在Uber应用中继续使用此功能。
- Exclusive products not available - Uber is continuously creating experimental or promotional products that may not be available to request through the Uber API
- 额外服务不可用 - Uber每时每刻都在创造实验性或是推广性的产品服务，但是这些产品服务可能无法通过Uber API来发起行程。

GUIDELINES AND ASSETS
设计规范和资源

To help you create an application that has an intuitive and familiar user experience, we have provided design assets, design guidelines, and an overview of how an entire Request can be implemented within an app.
为了帮助你开发一个拥有直观而熟悉的用户体验的应用程序，我们提供了关于如何将一个完整的行程请求植入到一个应用程序当中所需的设计资源、规范、概览（ design assets, design guidelines, and an overview）。