TOMP- API v2.0

Blueprint 2-wheeled shared vehicles

[Foreword 2](#_Toc202865683)

[Objectives 2](#_Toc202865684)

[Summary 3](#_Toc202865685)

[Flow overview 3](#_Toc202865686)

[(Offer &) purchase flows 4](#_Toc202865687)

[Asset based 4](#_Toc202865688)

[Offer based 5](#_Toc202865689)

[Product based 6](#_Toc202865690)

[Purchase flavors 8](#_Toc202865691)

[Immediate confirmed 8](#_Toc202865692)

[Auto-confirm 9](#_Toc202865693)

[2-phase purchase 9](#_Toc202865694)

[Execution flows 10](#_Toc202865695)

[Support flows 12](#_Toc202865696)

[Payment flow 13](#_Toc202865697)

[Deposits 13](#_Toc202865698)

[Pay when finished 13](#_Toc202865699)

[Subscriptions 13](#_Toc202865700)

[After sales flow 14](#_Toc202865701)

[Redresses 14](#_Toc202865702)

[Special cases 14](#_Toc202865703)

[Copyrights 16](#_Toc202865704)

# Foreword

The aim of this document is to describe the usage for 2-wheeled vehicles of the TOMP-API. The TOMP-API provides a wholistic API for all modes, and requires profiles per mode to make it understandable and easy to implement.

# Objectives

This document is written for these audiences:

* Software architects
* Software developers

The aim is to describe the process served by the API, make it relatable to your business process and to make it implementable.

# Summary

The TOMP-API includes the complete process of the user journey on 2-wheels (or a chained trip containing a part on 2 wheels).

The process is divided into several modules, each has its own flow.

The **offer flow** is possible, but not very common. But when the operator wants to be integrated in chained trips (based on journey planning), it has to supply this functionality. The search for offers, based on traveler’s requirements.

The ***purchase flow*** for 2-wheeled vehicles has 3 different starting points:

1. Start with a vehicle on the map, on the street or in external data sources
2. Purchase an offers (based on journey planning, the previous module)
3. Purchase additional products (subscriptions, day-cards etc.).

It has 3 flavors:

1. Immediately confirmed, but with a ‘cancel window’ to rollback without financial consequences
2. A pending package, where it is required to execute a second step, to confirm the purchase (or roll back the purchase)
3. A pending package, that is ‘auto-confirmed’ after the expiry time. Before the expiry time it is allowed to roll back the purchase.

The ***execution flow*** is pretty unique for 2-wheeled vehicles. It is because the traveller is in control, instead of someone else. It contains (amongst others) the steps of starting, ending, pausing, resuming the vehicle.

This part describes also the assignment of assets, and ancillaries.

The ***support flow*** is needed if you want to support unusual situations, like flat tires. To implement this, the reseller needs to implement the *notification module,* since it uses call-backs.

The ***payment flow*** can be used to report the balance, but also to request deposits. Optionally, it can be used to request direct payments. This part can be found in the *after-sales module.*

There are a few other modules, like the *customer registration module*, the *pre-sales module*, the *technical module* and the *discovery module*, but these are less relevant from the process perspective.

# Modes of operation

Normally, operators of the 2-wheeled vehicles are delivering a free-floating service or a station-based service (or sometimes a variant like virtual station-based). All these modes of operation can be served by the TOMP-API, where the boundaries of the free-floating areas, no-go areas and the (non-)parking zones can be published together with the package of services that has been bought (or offered).

It is possible to purchase a package with only a starting location (and time), specify the return area(s) or station(s), extend usage time, communicate with the user (out-of-boundaries, nearly-end-of-time, etc) and work with different pricing schemes, even with prices that depend on the availability.

# Flow overview

The process of using 2-wheeled shared vehicles is usually covered by this state transition diagram.

Afbeelding met tekst, diagram, lijn, Plan

Door AI gegenereerde inhoud is mogelijk onjuist.

In each of the flows described below, we use a blue references in italics, they refer to the appendix, where more details are provided.

In short, you should ponder which flows you want to implement, probably not all are relevant.

# (Offer &) purchase flows

## Asset based

This is a pretty common scenario. You get data for available bikes, scooters, etc., from a source (could be NeTEx, GBFS, or another standard), or maybe it's just right in front of you, on the street. You provide the ID (you can find references using the datasources, or it's the visual ID on the vehicle), and the MP can request to use this asset (use-asset).

Afbeelding met Mobiele telefoon, gadget, Elektronisch apparaat, Communicatieapparaat

Door AI gegenereerde inhoud is mogelijk onjuist.

NeTEx

GBFS

The response comes back in either a PENDING or CONFIRMED state (check out the *purchase flows*). In both cases, the required resources are claimed.

Afbeelding met tekst, schermopname, lijn, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

This function allows to book an asset with only one call! It returns a purchased package with a reference (deeplink) into your app, making it very easy to integrate into other solutions.

## Offer based

Another option is to search for offers based on start location, timestamp, and other user requirements (search-offers). These offers are NON-BINDING, meaning the resources aren’t claimed yet.  
  
You can then select a single offer and purchase it (purchase-offers).

Afbeelding met tekst, schermopname, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

The response from the purchase-offers call will be in either a **PENDING** or **CONFIRMED** state (see *purchase flows*). In both cases, the necessary resources are claimed.

Afbeelding met tekst, Mobiele telefoon, gadget, Draagbaar communicatietoestel

Door AI gegenereerde inhoud is mogelijk onjuist.If you’re allowed to modify offers (common in things like public transport or shared cars), or if you need to provide binding offers, you can use the pre-sales module.

However, since this isn't common in this context, it’s not covered in this blueprint.

## Product based

Afbeelding met tekst, schermopname, Lettertype, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Often, bike operators sell products like weekly cards, day cards or cards with e.g. 10 rides. These products can be sold using purchase-product.

In case of the 2-wheeled vehicles, it is also possible to purchase an ‘asset type’, like a family bike. This is also a product.

Afbeelding met tekst, gadget, Mobiele telefoon, Draagbaar communicatietoestel

Door AI gegenereerde inhoud is mogelijk onjuist.

The returned package is default in a CONFIRMED state, but it could be PENDING depending on the business case (see *purchase flows*). These purchased products normally don’t have consequences for claiming resources, but can later on be used in the offer flows (asset based or offer based), as cards.

# Afbeelding met tekst, Mobiele telefoon, gadget, Communicatieapparaat Door AI gegenereerde inhoud is mogelijk onjuist.Purchase flows

At this stage, a service in the shape of an asset or product has been selected (the state is PENDING or CONFIRMED). We are facing 3 different flows:

* immediate confirmed
* auto confirm and
* the 2-phase purchase

## Immediate confirmed

The returned purchase has the status ‘CONFIRMED’. Until the ‘Expiry-date’, the purchase can be undone using the rollback-purchase. This is a requirement, although the TO can specify itself how long it will give the opportunity to roll back the purchase without financial consequences.

After this timestamp, it can only be done using the refund-options, or in case of a technical problem, using cancel-package.

Afbeelding met tekst, schermopname, lijn, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

## Auto-confirm

The auto-confirm flow is more or less the same as the ‘immediate confirmed’-flow, except that the returned state is ‘PENDING’ until the timestamp specified in the expiry-date field has passed.

Afbeelding met tekst, lijn, schermopname, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

In this flow, it is also possible to request an extension of the expiry time.

## 2-phase purchase

Afbeelding met schermopname, tekst, lijn, diagram

Door AI gegenereerde inhoud is mogelijk onjuist.

The 2-phase purchase starts with a package in the PENDING state. The purchase-offers, use-asset and purchase-product, the returned information contains an expiration date.

If the PENDING package is not confirmed (confirm-purchase) and there is no request to extend the expiration time, it will automatically end in an EXPIRED state.

If the MP (on behalf of the customer) does not want to continue with the purchase process, it can roll back the purchase (rollback-purchase), thereby releasing the required resources.

# Execution flows

We have two levels of flows here. One on the package level, and the other on the leg level. The package level flow is straight forward, you have a package, you can modify the package (like assigning assets, or add ancillaries like helmets), and off you go.

At the end, when all legs are ended, the package ends up in the state ENDED.

Afbeelding met tekst, diagram, schermopname, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

At leg level, there are more options. These leg flows can only be activated when the package is in STARTED state.

When the leg is in NOT STARTED state, it is possible to assign assets and ancillaries. This could impact the total price.

Afbeelding met tekst, schermopname, Lettertype, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.Once a TO gets a request to start a leg (start-leg), it is possible that the TO needs some time to prepare the applicable asset related to the leg. It will return a leg in the state ‘PREPARING’. For this flow, the *notification module* must be implemented at the MP side, so the TO can inform the MP that the asset is prepared.

Afbeelding met tekst, elektronica, gadget, Elektronisch apparaat

Door AI gegenereerde inhoud is mogelijk onjuist.But, in most cases, when starting a leg, it will return a leg ‘in use’. The asset is unlocked and ready to use. In some occasions, the asset is not unlocked, but must be unlocked manually.

To perform operations directly on the asset, there is an operation-asset facility, to e.g. open a trunk, or to unlock a side door in a parking garage. This is not described in this blueprint, but it’s there.

Once the asset is in use, it can be paused (pause-leg) and resumed (resume-leg) again, or when needed, and the end time of leg is approaching (extend-leg), you can request to extend the leg as traveller.

Afbeelding met tekst, diagram, schermopname, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.Ending the leg (end-leg) implies that the asset is locked and the financial calculations can be made.

# Afbeelding met tekst, persoon, gadget, Mobiele telefoon Door AI gegenereerde inhoud is mogelijk onjuist.Support flows

The support flow can only be activated per leg, when it is in execution. From any state within the leg-flow, an issue can be reported, and it has to be handled appropriately by the TO.

The MP is of course in charge of issues related to the app and the communication between the TO and MP, but when it relates directly to the asset, the MP has to rely on the TO to fix the issue.

Afbeelding met tekst, schermopname, diagram, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.The help desk or support officer from the TO has multiple options, like fixing things at location, assign another asset to the leg (e.g. a replacing bike) or request an abnormal end of the leg. In this case, the after sales module (requiring refunds) facilitates the process.

# Payment flow

In this context, it is likely that the TO uses deposits, a kind of financial guarantee. For instance, when the MP and TO don’t know each other, it is a normal way of working.

## Deposits

Afbeelding met tekst, diagram, schermopname, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Once the payment has been confirmed (and the TO can of course validate it on its bank account), the purchase can be returned. This approach can also been taken when the MP does not want to pay the deposit; it can relay the payment request to the customer.

This approach requires the request-payment (in the notification module) and the confirm-payment.

## Pay when finished

Beside upfront payments and deposits, we also know organizations that request immediate payment when the leg has ended. The same approach can be applied.

## Subscriptions

The majority of the implementations uses nowadays subscriptions. It is required to publish somehow, as a TO, how much credits you have from a certain MP. The payments can be used for this, so the MP can validate their own registration with the TOs.

# Afbeelding met tekst, schermopname, Lettertype, lijn Door AI gegenereerde inhoud is mogelijk onjuist.After sales flow

## Redresses

The after sales module contains, besides the payments, also the redress option. A redress can be a refund (‘money back’) or a replacement (another package, maybe valid on another day, or a week-card, same trip somewhat later including a cup of coffee, … )

Supplying the legs, ancillaries or package you want to request redresses for, will return redress options. If applicable, the TO returns valid redress options, it can contain financial compensations or replacements.

# Special cases

There are a few special cases (in comparison to other modes) in this blue print:

* blue tooth locks
* manual locks
* (non)-parking and other zones
* return areas or return stations
* open helmet-boxes
* communication of instructions
* warnings to the traveller

We’ll describe these in just a few lines per case.

## Blue tooth locks

Afbeelding met tekst, gadget, Mobiele telefoon, fiets

Door AI gegenereerde inhoud is mogelijk onjuist.Blue tooth locks are not standardized at all. This makes it hard to communicate with them. We see things like SDKs, but this limits the usage to the apps who want to incorporate the SDKs. And if you want to use many of these, your MP app will become very hard to maintain and very big. We don’t have a straightforward answer to this one.

## Manual locks

These locks require instructions, and those can be supplied using the ‘links’ part of the leg. A simple leg referencing to an instruction page on the internet, with a ‘required’ : true field, would force the MP to show the instructions. Often, the bike will communicate to its back-office that it has been opened, which can trigger a notification to the MP.

## Afbeelding met tekst, Mobiele telefoon, gadget, Elektronisch apparaat Door AI gegenereerde inhoud is mogelijk onjuist.(Non)-parking and other zones

During the ‘execution’ phase, the operator can choose to deliver geoJSON, including features referencing to e.g. MDS policy areas.

## Return areas or return stations

The same applies to return areas or stations, the stations or return areas can be represented as features, referring to GBFS items.

## Open helmet-boxes / custom asset operations

It is possible and allowed to ‘invent’ new endpoints, as long they comply to OGC processes. The flow can be altered, because you – as TO – prescribe the following steps in each returned concept (like a leg).

This could result in a POST /processes/open-helmet-box/execute, and the start-leg could specify a ‘open-helmet-box’ link. When this URL is called, the helmet box opens and in the result there is a ‘unlock-asset’ link and optionally a link with instructions.

## Afbeelding met tekst, motorfiets Door AI gegenereerde inhoud is mogelijk onjuist.Communication of instructions

In e.g. scooter solutions, it is very common to give instructions to operate the scooter, before each step. It is possible to add ‘instruction’ links in the link set, and you could even mark them as ‘mandatory’, to indicate the MP to show the website (or any other link) before the next link is executed.

Afbeelding met tekst, gadget, fiets, Draagbaar communicatietoestel

Door AI gegenereerde inhoud is mogelijk onjuist.

## Warnings to the traveller

This functionality must be implemented by the MP. It can be found in the notification module, and the TO can send notifications of the type WARNING or INFORMATION (or even more detailed notifications). The destination should be ‘TRAVELLER’.

# Copyrights

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at  
  
 http://www.apache.org/licenses/LICENSE-2.0  
  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.