PITANE MOBILITY

OPEN API VERSION 3

REST SERVER

PITANE B.V.

Pitane Mobility API – V3

Development Guide for Pitane Mobility – Pitane B.V.

TABLE OF CONTENT

VERSIONS AND PROTOCOL

When changes are made to the Pitane Mobility Rest Server protocol, these are often fields that are added to a specific message. When this is the case, the version number is placed behind this new field in which this field first appeared. The basic functions that are in version 2.0.0 are not described because otherwise it is no longer clear to see which changes have been added in a new protocol version. This guide describes some common practices useful for setting up your web service requests and processing your web service responses. HTTPS is required for all Pitane Mobility JSON API web service requests containing user data, or developer identifiers. Requests made over HTTP that include sensitive data may be rejected.

BUILDING A VALID URL

You may think that a "valid" URL is self-evident, but that's not quite the case. A URL entered within an address bar in a browser, for example, may contain special characters (e.g. "éęėæãaü"); the browser needs to internally translate those characters into a different encoding before transmission. By the same token, any code that generates or accepts UTF-8 input might treat URLs with UTF-8 characters as "valid” but would also need to translate those characters before sending them out to a web server. This process is called URL-encoding. We need to translate special characters because all URLs need to conform to the syntax specified by the W3 Uniform Resource Identifier specification. In effect, this means that URLs must contain only a special subset of ASCII characters: the familiar alphanumeric symbols, and some reserved characters for use as control characters within URLs.

LIABILITY

Pitane BV and / or Pitane Mobility do not accept any responsibility or consequential damage in the use of links in combination with this protocol. If it turns out that integrated components no longer work as a result of an amendment to our protocol, no responsibility or consequential damage will be accepted. We always advise you to consult the changes in advance to check the possible impact of the changed implementation. By using the Pitane Mobility Rest Server, users accept the risks that technical failures can occur.

REST-COMMUNICATION

REST stands for Representational State Transfer. It is a software architecture style that relies on a stateless communications protocol, most commonly, HTTP. REST structures data in XML, YAML, or any other format that is machine-readable, but usually JSON is most widely used. REST follows the object-oriented programming paradigm of noun-verb. REST is very data-driven, compared to SOAP, which is strongly function-driven. You may see people refer to them as RESTful APIs or RESTful web services. They mean the same thing and can be interchangeable. There is no standard for the description format of REST services. The Pitane Mobility API typically have a request-response structure. Input is sent in as URL parameters or in the body of the request to the API endpoint, the business process engine executes the functionality based on the inputs and responded with XML or JSON datatypes.

OUR API TECHNOLOGY

In our production environment there is a load balancer that distributes incoming traffic evenly over different servers. The Pitane Mobility API is always available on multiple physical machines divided over two availability zones in Europe. This makes it possible for our services to operate data center-redundantly. For the time being, we do not have any limitation on data traffic to the Pitane Mobility API, but for requesting a position of a vehicle, for example, we recommend a minimum of 60 seconds interval because the car itself also offers a new position every 60 seconds. Multiple same requests within a minute are a waste of the requests. We assume that the speed of our Pitane Mobility API means that most functions can be called on-demand and that request timers are not necessary. Access key All calls to the Pitane Mobility API must include a client key, please keep this private because anyone with this key can access and or modify your data. It is prohibited to use tools for collecting data and/or storing data in local or own databases. Every call must be originating from the phone, tablet or personal device of the user who's the personal owner of the data. This regarding GDPR or AVG regulations.

General quota limit

While you are no longer limited to a maximum number of requests per day (QPD), the following usage limits are still in place for the Pitane Mobility API:

- 50 requests per minute (QPM) per IP address. This quota is referred to as **loads per minute per user**. By default, this quota is set to 50 requests per 60 seconds per user and can be adjusted to a **maximum** value of 1,000. Number of requests to the Pitane Mobility API is restricted to a **maximum** of 10 requests per second per user.
- If **you exceed** this amount, subsequent requests to the **API** will fail with the OVER_QUERY_LIMIT status code and the data will not be returned.
- MAC address spoofing, multiple VPN connections or DoS attacks may result in blacklisting the client without any message prior to blocking.

Retry mechanism

If you have an endpoint method that fails occasionally, you need to retry it a few times before throwing an exception. Call the Pitane Mobility API with a method call that will throw an exception only after three failed executions with a 5 seconds interval between them.

As a general rule, 500 errors should be retried. There is a rule of thumb for most error codes:

- 2xx = Everything that can be done has been done.
- 4xx = You (the caller) did something wrong or asked for something nonsensical.
- 5xx = We (the server) did something wrong.

4xx errors generally boil down to a request that is inherently bad and cannot be handled. 5xx errors are used to indicate that the server encountered a private issue, and there is no indication that your request is provably invalid.

That doesn't mean your request must have been valid. For example, let's say you send a request to fetch an item but you use an ID that doesn't exist. A server that handles the request well will give you a 404 response (or a 204. I've heard arguments either way and I think the distinction is contextual). However, if the server has a bug which leads to a null reference exception, you're likely to get a 500 response.

As the caller, when you get a 500 response, you don't know what went wrong. All you know is that the server did not actively tell you that your request is nonsensical. Which means that you cannot assume that there's no point in trying to make the same request again. For example, maybe you received the 500 because the server database was offline. If you wait a bit and try again, it may succeed now that the database is online.

But when you receive a 4xx error, the server has actively stated that your request is unresolvable and there's no point in retrying.

This is a bit of an overgeneralization, fringe exceptions exist. For example, status 429 (too many requests) means that you (the caller) have exceeded your allotted amount of calls, but it's likely that your same request will be processed correctly if you wait until you've been allotted more requests (e.g. if you hit the daily maximum, it will work again tomorrow).

DEVELOPMENT TIPS

We want third parties and developers to be assisted in any way. For that reason, most of the functions can be tested online on your live-data at accept.pitane.dev. Use your personal API key and be carefully since this is production data regulated by GDPR and AVG. We always suggest using the test environment during development. All fields in de data section of this API are well documented in the SQL SERVER DATA MODEL. Use this documentation to have detailed information on field names, types, lengths and data representation. This additional manual is available for all our customers, software houses and developers integrating personal systems on the Pitane Mobility network.

Our **open API** (often referred to as a public API) is a publicly available application programming interface that provides developers with programmatic access to a proprietary software application or web service.

Test environment:

Server: accept.pitane.dev
Port: 80/443
Protocol: HTTP/HTTPS
Access key (DEMO): bd7f7507-7c49-475c-9281-867e4a35b244

Production environment:

The production key will be delivered after following criteria are successfully completed:

- The customer and/or owner of the data agreed to deliver a production key
- The customer signed the 'VERWERKERSOVEREENKOMST' document
- The application has been tested with success in a sandbox environment
- The third party signed the 'SUB-VERWERKERSOVEREENKOMST' document

Any additional cost for consumer keys in the application of the third party like Google Maps platform, Flight Aware or SMS hosting providers are the responsibility of the customer and/or owner or third party. Production keys could change, and production servers could be on any location in Europe or any IP address in our datacenter. Before any changes an announcement will be done the involved parties.

Architecture key Points:

REST is an architecture (or even better, an architectural style) it is clearly not a standard, although it uses several existing standards like HTTP, URL, plus many format types for the actual data.

- REST uses HTTP methods to indicate which operation to perform (retrieve or HTTP GET, create or HTTP PUT, update or HTTP POST, and delete or HTTP DELETE)
- REST uses HTTP parameters (both as query parameters and POST parameters) to provide further information to the server
- REST relies on HTTP for authentication, encryption, security (using HTTPS)
- REST returns data as plain documents, using multiple mime formats (XML, JSON, images, and many others)

Default production server:

Server: api.[server].[ext]
Port: 80/443
Protocol: HTTP/HTTPS
Access key: [request production key]

SSL, IDENTITY, CONTENT VALIDATION AND PRIVACY

We integrate or implement key-based authorization to use the API. This is commonly used to authorize mobile applications to use web APIs. One of the most common problems with API keys are their protection. You will need to find a way to make it reasonably difficult for anybody to steal those keys. Ultimately this is often impossible with mobile applications, but you always need to make the risk evaluation anyway and see what reasonable mitigation for your found risks is. Pitane Mobility allows system managers to generate API user's rights on any API endpoint.

Different applications should use different API keys to prevent the system. Always take moment to evaluate the risks and the value of whatever you are trying to protect. Then you will know what the reasonable cost and effort is to use for the protection. Without API Security, enterprise core functions are at risk of data theft and disruption. IT professionals, regardless of their function, should keep a close eye on building the foundations for their infrastructure on the 4 tenants of API security: SSL, Identity, Content Validation and Centralized Enforcement.

PITANE MOBILITY API PRIVACY

You may provide personal information to us through the Pitane Mobility API Services – for example, when you create an account, contact customer support, send us an email, or communicate with us in any other way. When setting up an account, you will be asked to provide certain basic information such as a name, email address, username, password, company name, location and phone number. You may also need to provide us with payment and billing information such as the customer's credit card details and billing address. As a customer, we will also maintain a record of your purchases, transactional information, services history and usage, and any communications and responses. We may also collect personal information, such as your contact information and feedback.

INFORMATION WE COLLECT

When you use the API services, we may collect certain information automatically about your device and your use of the services. This will include IP addresses, browser profiles (user agents), log files, and other information regarding your system and connection. We collect information about how you access and use the services, such as what pages are viewed and what portions of the services are used.

We also collect information regarding the performance of the services, including metrics related to the uptime of hosted services. This information allows us to improve the content and operation of the services and facilitate research and analysis of the services.

We collect this information automatically through the use of various commonly used information-gathering technologies including cookies and web beacons, to collect information as users and visitors navigate the website and use the API services ("Web Site Navigational Information").

We use these technologies to analyze trends, administer web sites and services, track users' and visitors' movements around our website and services, and gather demographic information about our user and visitor base as a whole.

We reserve the right to change or update this policy from time to time. If material changes are made, we will place a prominent notice on our website or services or documentation for at least 30 days prior to the change taking effect or communicate with you directly by email or through the services and will update the last revised date at the top of this policy. We encourage you to regularly check back on this page to ensure you are up to date with any changes.

REQUEST YOUR PERSONAL TOKEN

If you wish to use our Pitane Mobility Rest network, you must have a valid token or API. You can request this via helpdesk@pitane.nl or receive the key directly from the company with whom you wish to build up the communication.

DATA TYPES AND STANDARD VALUES

STRING or VARCHAR	ASCII-value 0-9/ a-Z – ABC-12abcëô
BOOLEAN (false/ true	0 of 1
DATE	jjjj-mm – 2016-12-
DATETIME	jjjj-mm-ddTHH:nn:ss – 2016-12-24T12:34:00
INTEGER	Number- 12345
FLOAT	Number with decimal point – 123.45
DECIMAL	Number with decimal point – 123.45
EXTENDED	Number with decimal point – 123.45
LATITUDE	Number with decimal point – 51.1234567
LONGITUDE	Number with decimal point – 5.1234567

When building the XML or JSON, all values must be filled even if they are not known to the host, except for a STRING OR VARCHAR value. All non-INTEGER numeric values must contain decimal point.

Next values are the default values for the field types

INTEGER	0
DATE/TIME	1900-01-01T00:00:00 (Central European Time)
BOOLEAN	0
FLOAT	0.0
DECIMAL	0.0
EXTENDED	0.0
LATITUDE	0.0
LONGITUDE	0.0

MESSAGE COMPOSITION

The results consist of several parts. The first part is the header that is equal for each package in terms of structure. Below are the details and these are could be different for each JSON/XML request. The header always consists of the same number of tags. Below is a summary of the header tags and an example as it appears in the JSON/XML.

Tag	Remark	Sample
key	Personal key or API	[your API key]
output	Json(default) or XML generated output	json
method	GET / POST / PUT / DELETE	GET
Ip	IP address host	1854.256.256.17
status	Status of the result	OK
records	Number of received records	0 -1000
remark	Remark or extra notification	Voorbeeld: FFFFFFFFFF
request	Endpoint that has been requested	pitaneCompaniesRetrieve
key-tag	Key used fort his API request	wag_id
build	Build version API	2.0.0.143
copyright	Copyright information	Pitane BV
servertime	Actual server timestamp	2018-06-24T12:34:14
serverseconds	Date/Time prior to January 1, 1904	3611302261.0
Optional	*Optional parameters during request*	
Details	*Echo the parameters*	

SAMPLE URL JSON/XML

JSON:

/pitane/rest/TPV3/Service/[KEY]/json/pitaneDataAirportsCartypes?air_iata=AMS

XML:

/pitane/rest/TPV3/Service/[KEY]/xml/pitaneDataAirportsCartypes?air_iata=AMS

Additional parameters:

- To return parameters in detail add: &detail=1

AVAILALBLE ENDPOINTS

Pitane Mobility REST Server – Open API v3.x

Endpoints could be categorized by service type.

https://app.swaggerhub.com/apis-docs/Pitane-Mobility/Pitane-API/3.0#/

More information and samples are available on SwaggerHub.

This platform an integrated API development platform that brings together all the core capabilities of our open source framework, along with additional advanced capabilities to build, document, manage, and test Pitane Mobility API's.

ERRORS / STATUS

The Pitane API reports internal errors as a separated JSON or XML block. Please see HTTP status codes for more information about the errorstatuscode.

ERRORS / STATUS CODES

Status	Description	Remark
200	OK	
401	Not found	
405	Method not allowed	
406	Not acceptable	
409	Conflict	
501	Not implemented	

Sample errorstatus

```
<xml>
<servertime>2020-07-08T12:41:37:737</servertime>
<key>bd7f7507-7c49-475c-9281-867e4a35b244</key>
<output>xml</output>
<request>pitaneDeviceSendServiceStart</request>
<protocol>HTTP/1.1</protocol>
<method>GET</method>
<appname>Pitane Driver</appname>
<ip>136.144.238.137</ip>
<remoteaddr/>
<owner>Taxicentrale & Zorgvervoer</owner>
<remark/>
<copyright>Pitane B.V.</copyright>
<error>
        <errorstatus>406</errorstatus>
        <errormessage>Driver or vehicle already in service</errormessage>
</error>
<version>3.0</version>
<build>3.0.0.3</build>
<status>OK</status>
</xml>
```

TRANSACTIONS

Database inserts, deleted or updates are responded by transaction block. Please see tran

Sample XML TRANSACTION BLOCK

```
<transaction>
	<transactionstatus>202</transactionstatus>
	<transactionmessage>UPDATED</transactionmessage>
</transaction>
```

TRANSACTION CODES

Status	*Description*	*Remark*
200	OK	
201	INSERTED	
202	UPDATED	
203	DATA NOT ACCEPTED	
204	FIELDS INCOMPLETE	
205	SELECTED	
409	CONFLICT	

SAMPLES

UPDATE RECORD

```
<xml>
	<servertime>2020-07-08T09:21:18:718</servertime>
	<key>bd7f7507-7c49-475c-9281-867e4a35b244</key>
	<output>xml</output>
	<request>pitaneSendTripAssigned</request>
	<protocol>HTTP/1.1</protocol>
	<method>GET</method>
	<appname>Pitane Driver</appname>
	<ip>136.144.238.137</ip>
	<remoteaddr/>
	<owner>Taxicentrale & Zorgvervoer</owner>
	<records>0</records>
	<remark/>
	<copyright>Pitane B.V.</copyright>
	<transaction>
		<transactionstatus>202</transactionstatus>
		<transactionmessage>UPDATED</transactionmessage>
	</transaction>
	<version>3.0</version>
	<build>3.0.0.3</build>
	<status>OK</status>
</xml>
```