

Paris Airports - Web API Flight Informations



AÉROPORTS DE PARIS

Flight Information Hackathon – A660

Version			
Version	Date	writer	Comment
1.0	11/04/2015	Olivier MONGIN	Document creation
1.1	12/04/2015	Olivier GOSSEAUME	Document revision
1.2	24/04/2015	Olivier GOSSEAUME	Document revision
1.3	06/05/2015	Olivier GOSSEAUME	Document revision
1.4	21/05/2015	Olivier GOSSEAUME	Document revision (examples added)
1.5	29/05/2015	Olivier MERCIER	Document revision and validation

Rédacteur : GOSSEAUME Olivier	Date : 11/04/2015
Approbateur : MERCIER Olivier	Date : 29/05/2015

SOMMAIRE

1	INTRODUCTION	4
1.1	Protocol	4
1.2	On line help	4
2	REST SERVICES	4
2.1	API key	4
2.2	Resources	4
2.2.1	Countries	5
2.2.2	Aircrafts types	7
2.2.3	Aircrafts registrations	9
2.2.4	Airlines	11
2.2.5	Terminals	13
2.2.6	Airports	15
2.2.7	Flights	17
3	REQUESTS EXAMPLES	30

1 Introduction

This web API is a RESTful type Web API. It provides real time information about Flights that arrive at or depart from both Paris airports (Paris Orly and Roissy CDG). You can also obtain all reference data pertaining to flights (airlines, airports and countries, aircrafts types and registration, terminals).

The time period you can query for flights span from 2 days in the past to 30 days in the future:

- the [Day-2, Day+2] period is updated in near real time (every 30 seconds approx.)
- the [Day+3, Day+30] period is updated less frequently (every 30 minutes approx.) because it contains only scheduled flight information.

1.1 Protocol

This API use HTTP protocol. You can decide to use JSON or XML. If you are building Web Pages for AJAX calls, then JSON is a good choice.

To query the API you will have to use an API key that will be given to you in order to authenticate you against the server.

1.2 On line help

This API provides online help pages to help you use it. The URL is here :

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/help>

2 REST services

2.1 API key

To be able to call a resource URI you will need to provide an API key in parameter "apiKey". This key is an authentication key that will be provided to you at the beginning of the hackathon. The same key can be used with all resources.

2.2 Resources

2.2.1 Countries

Get information about countries either as a list or by country code (one occurrence).

2.2.1.1 Country description

Describes a country :

Name	Type	Description
CountryCode	string	ISO country code
NameFR	string	Name in french
NameEN	string	Name in english (if available)
CustomsStatus	string	Customs controls applicable at Paris for this country ("international" means mandatory control, "schengen" means control for non Schengen citizens entering Schengen zone, "domestic" means domestic flight in France)

2.2.1.2 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Countries?apiKey={apiKey}</i>	HTTP/1.1
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Countries/{id}?apiKey={apiKey}</i>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	CountryCode

2.2.1.3 Response

Response body:

```
{
  "countryCode": "FR",
  "nameFR": "FRANCE",
  "nameEN": "FRANCE",
  "customsStatus": "domestic"
},
```

```
{  
  "countryCode": "GA",  
  "nameFR": "GABON",  
  "nameEN": "GABON",  
  "customsStatus": "international"  
},  
{  
  "countryCode": "GB",  
  "nameFR": "ROYAUME-UNI",  
  "nameEN": "UNITED KINGDOM",  
  "customsStatus": "domestic"  
}
```

2.2.2 Aircrafts types

Get information about aircrafts types either as a list or by aircraft type code (one occurrence).

2.2.2.1 Aircraft description

Describes an aircraft type, ie a plane doing flights:

Name	Type	Description
AircraftCode	string	Aircraft code (which is a IATA code or IATA sub-code)
IATACode	string	IATA code for aircraft type
IATASubCode	string	IATA sub-code for aircraft type (ie more precise code)
ICAOCode	string	ICAO code for aircraft type
ShortNameFR	string	Short name in french
LongNameFR	string	Long name in french
WakeCategory	string	Weight or Wake category ("light" or "medium" or "heavy")
EngineType	string	Engine type ("jet" or "piston" or "helicopter" or "turbo prop")

2.2.2.2 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<code>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Aircrafts?apiKey={apiKey}</code>	HTTP/1.1
GET	<code>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Aircrafts/{id}?apiKey={apiKey}</code>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	AircraftCode

2.2.2.3 Response

Response body:

```
{
  "aircraftCode": "38F",
  "iataCode": "38F",
  "iataSubCode": "38F",
```

```
"icaoCode": "A388",  
"shortNameFR": "A380-800F FRT",  
"longNameFR": "AIRBUS INDUSTRIE A380-800F FREIGHTER",  
"wakeCategory": "heavy"  
},  
{  
  "aircraftCode": "380",  
  "iataCode": "380",  
  "shortNameFR": "A380 PAX",  
  "longNameFR": "AIRBUS INDUSTRIE A380 PASSENGER",  
  "wakeCategory": "heavy"  
}
```


2.2.3 Aircrafts registrations

Get information about aircrafts registration either as a list or by aircraft registration code (one occurrence).

2.2.3.1 Registration description

Describes an aircraft registration (ie an unique alphanumeric string that identifies a civil aircraft, in similar fashion to a license plate on an automobile):

Name	Type	Description
RegistrationCode	string	Registration code
AircraftCode	string	Aircraft code associated with this registration (see "aircrafts" resource)
OwnerAirlineCode	string	Airline owning this aircraft (see "airlines" resource)
UserAirlineCode	string	Airline usually using this aircraft for flight operations (see "airlines" resource)

2.2.3.2 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Registrations?apiKey={apiKey}</i>	HTTP/1.1
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Registrations/{id}?apiKey={apiKey}</i>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	RegistrationCode

2.2.3.3 Response

Response body:

```
{
  "registrationCode": "BA6WX",
  "aircraftCode": "AR8",
  "ownerAirlineCode": "AF",
  "userAirlineCode": "AF"
}
```

```
},  
{  
  "registrationCode": "BBIJZ",  
  "aircraftCode": "320",  
  "ownerAirlineCode": "AF"  
},  
{  
  "registrationCode": "BCJKH",  
  "aircraftCode": "EMB",  
  "ownerAirlineCode": "SWT",  
  "userAirlineCode": "FDX"  
}
```

2.2.4 Airlines

Get information about airlines either as a list or by airline code (one occurrence)

2.2.4.1 Airline description

Describes an airline (ie a flight carrier):

Name	Type	Description
AirlineCode	string	Airline code (which is a IATA code or ICAO code)
IATACode	string	IATA code for airline (if available, not all airlines have a IATA code)
ICAOCode	string	ICAO code for this airline (if available)
AirportCode	string	Basement airport code (if available, see "airports" resource)
CountryCode	string	Country code (see "countries" resource)
NameFR	string	Name in french
NameEN	string	Name in english (if available)

2.2.4.2 Request

You can use this API with HTTP GET.

Request URI	Version
GET <i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Airlines?apiKey={apiKey}</i>	HTTP/1.1
GET <i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Airlines/{id}?apiKey={apiKey}</i>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	AirlineCode

2.2.4.3 Response

Response body:

```
{
  "airlineCode": "AF",
  "iataCode": "AF",
  "icaoCode": "AFR",
  "airportCode": "CDG",
  "countryCode": "FR",
  "nameFR": "AIR FRANCE",
```

```
"nameEN": "AIR FRANCE"  
},  
{  
  "airlineCode": "KL",  
  "iataCode": "KL",  
  "icaoCode": "KLM",  
  "airportCode": "AMS",  
  "countryCode": "NL",  
  "nameFR": "KLM",  
  "nameEN": "KLM"  
}
```

2.2.5 Terminals

Get information about terminals either as a list or by terminal code (one occurrence).

2.2.5.1 Terminal description

Describes a terminal in Paris (Orly or Roissy CDG):

Name	Type	Description
TerminalCode	string	Terminal code
AirportCode	string	Airport code (see "airports" resource). Either ORY or CDG
Name	string	Name of the terminal

2.2.5.2 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Terminals?apiKey={apiKey}</i>	HTTP/1.1
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Terminals/{id}?apiKey={apiKey}</i>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	TerminalCode

2.2.5.3 Response

Response body:

```
{
  "terminalCode": "C2",
  "airportCode": "CDG",
  "name": "CDG 2"
},
{
  "terminalCode": "C2A",
  "airportCode": "CDG",
  "name": "CDG 2 A"
},
{
  "terminalCode": "C2B",
  "airportCode": "CDG",
  "name": "CDG 2 B"
}
```


}

2.2.6 Airports

Get informations about airports either as a list or by airport code (one occurrence).

2.2.6.1 Airport description :

Describes an airport or a city containing airports :

Name	Type	Description
AirportCode	string	IATA code of this airport
ICAOAirportCode	string	ICAO code of this airport (if available)
CityAirportCode	string	ISO code of city containing this airport (a city is also registered as an airport)
CountryCode	string	ISO country code (see "countries" resource)
NameFR	string	Name in french
NameEN	string	Name in english (if available)
Latitude	string	WGS84 Latitude (if available)
Longitude	string	WGS84 Longitude (if available)
Altitude	string	Altitude in meters (if available)
OrthoDistance	string	Approx. orthodromic distance in kilometers from Paris (if available)
FlightTimeFromParis	string	Approx. time to flight in minutes from Paris (if available)
FlightTimeToParis	string	Approx. time to flight in minutes to Paris (if available)
TimeZoneCurrent	string	Current time zone taking in account daylight saving time as of today in format UTC+HHMM or UTC-HHMM (HH for hours, MM for minutes)

2.2.6.2 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<code>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Airports?apiKey={apiKey}</code>	HTTP/1.1
GET	<code>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Airports/{id}?apiKey={apiKey}</code>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	AirportCode

2.2.6.3 Response

Response body:

```
{
  "airportCode": "AMS",
  "icaoAirportCode": "EHAM",
  "countryCode": "NL",
  "nameFR": "AMSTERDAM",
  "nameEN": "AMSTERDAM",
  "latitude": 52.308613,
  "longitude": 4.763889,
  "altitude": -3,
  "orthoDistance": 430,
  "flightTimeFromParis": 70,
  "flightTimeToParis": 65,
  "timeZoneCurrent": "UTC+0200"
},
{
  "airportCode": "ORY",
  "icaoAirportCode": "LFPO",
  "cityAirportCode": "PAR",
  "countryCode": "FR",
  "nameFR": "PARIS ORLY",
  "nameEN": "PARIS ORLY",
  "latitude": 48.725278,
  "longitude": 2.359444,
  "altitude": 89,
  "orthoDistance": 11,
  "flightTimeFromParis": 45,
  "flightTimeToParis": 45,
  "timeZoneCurrent": "UTC+0200"
}
```

2.2.7 Flights

Get information about flights either as a list (with criterions) or by flight id (one occurrence).

2.2.7.1 Flight description

Describes a flight, ie a plane carrying passengers going to Paris (arrival flight) or coming from Paris (departure flight). If a flight is only transiting in Paris you will get an arrival and a departure flight :

Name	Type	Description
Ident	FlightDesc	Flight identification descriptor
AirportCode	string	For a departure flight, gives you the departing airport (either ORY for Orly, or CDG for Roissy). For an arrival flight, gives you the arrival airport (either ORY for Orly, or CDG for Roissy)
ArrDep	string	Arrival (A) or Departure (D) flight
JointAirlineCode	string	Secondary (joint) airline Code (see "airlines" resource). A possible second airline operating this flight
FlightDate	date	Scheduled date of flight (local time)
FlightStatus	string	Flight status. Possible values : "active" : the flight is active, "cancelled" : the flight have been cancelled, "inactive" : the flight is not active anymore because it is either reported to another day or diverted
Operating	FlightDesc	Operating flight information. Some airlines (in alliances for example) can share the same physical plane. This is called code sharing. Thus a physical plane can transport passengers with different flight numbers. One and only one commercial flight is the "operating" (the master flight), and all others are "marketing" (the slaves flights). This field give you the operating flight if the current flight is a marketing one.
Onward	FlightDesc	Onward flight information. For an arrival, it gives the departure flight that will use the same physical plane (same registration). For a departure, it gives the arrival
OriginAirportCode	string	Airport code of origin airport (ie from

		where the flight is coming from)
OriginFlightDate	date	Scheduled date of origin airport (in local time at the origin airport)
DestinationAirportCode	string	Airport code of final destination airport (ie where the flight is going to)
DestinationFlightDate	date	Scheduled date of final destination airport (in local time at the destination airport)
ServiceType	string	IATA Service type. This gives information on passenger cabin configuration. Possible values : "J" - Scheduled passenger service, "F" - Scheduled cargo, "G" - Additional passenger scheduled service, "C" - Charter passenger service, "H" - Charter cargo, "P" - Positioning flight, "T" - Technical test, "K" - Training, "X" - Technical stop, "D" - General aviation,
TerminalCode	string	Terminal code (see "terminals" resource). This is the terminal in which the passengers will checkin (departure) or arrive (arrival)
AircraftCode	string	Aircraft type code (see "aircrafts" resource)
RegistrationCode	string	Aircraft registration code (see "registrations" resource)
ATCCallsign	string	ATC (Air Traffic Control) callsign of the flight. ATC does use this callsign to control air traffic. You can use this callsign to locate the flight on sites like "flightradar24.com"
RunwayCode	string	Runway threshold information for the flight, also called QFU. Each runway have two threshold (one at each end) which are painted on runway and visible on Google earth
ParkingCode	string	Parking stand information. Gives the parking code (visible on Google earth) where the plane is physically located or directed to
BusNeeded	boolean	True is passengers will embark or disembark with a bus. This is necessary where the aircraft parking stand does not touch the terminal with a jet bridge.
Customs	string	Customs controls status of the

		flight. Describes what kind of customs formality the passengers have to go thru. Possible values : "international" : full passport control needed, "schengen" : Schengen flight, passport control needed for non Schengen citizens entering Schengen zone, "domestic" : domestic flight, no passport control needed
LastUpdate	date	Last time the flight was updated
FIDS	FIDSData	FIDS (Flight Information Display) information. This contains informations displayed on public screens in airport
Times	TimesData	Detailed time informations
Route	Collection ofRouteData	Flight routing informations (one entry per flight leg). First entry for an arrival flight is the origin airport. Last entry for a departure flight is the final destination airport
Checkin	CheckinData	Detailed passenger check-in informations (departure flight only)
Boarding	BoardingData	Detailed passenger boarding informations (departure flight only)
Arrival	ArrivalData	Detailed passenger arrival informations (arrival flight only)
Delivery	BaggageDeliveryData	Detailed baggage delivery informations (arrival flight only)

2.2.7.2 FlightDesc description

Identifies a flight descriptor. Concatenating AirlineCode and FlightNumber gives you the full flight number a passenger sees on its boarding pass :

Name	Type	Description
Id	Integer (long)	Opaque Id which identifies uniquely the flight in our systems (can be used to search by Id)
AirlineCode	string	Airline code (see "airlines" resource). This is the airline code a passenger sees on its boarding pass
FlightNumber	string	Flight number (3 to 4 digits, eventually followed by a letter). Concatenating AirlineCode and

		FlightNumber gives you the full flight number a passenger sees on its boarding pass
--	--	---

2.2.7.3 FIDSData description

FIDS (Flight Information Display) information. This contains informations displayed on public screens in airport :

Name	Type	Description
PrimaryStatus	string	Codified primary status of the flight (see also "PrimaryInfo"). Possible values : "on_time" : flight is expected on time, "expected" : flight is expected at "PrimaryData" time, "landed" : arrival flight have landed in Paris at "PrimaryData" time, "arrived" : arrival flight arrived at parking at "PrimaryData" time, "take_off" : departure flight had taken off at "PrimaryData" time, "waiting" : flight is waiting for further information, "boarding" : boarding in progress, "boarding_closed" : boarding completed, "cancelled" : flight is cancelled, "delayed" : flight is delayed, "diverted" : flight diverted to another airport. No further information, "diverted_to" : flight diverted to "PrimaryData" airport, "postponed" : flight is postponed to another date. No further information, "postponed_to" : flight is postponed to "PrimaryData" date and flight, "bag_return" : baggage return after late departure flight cancellation
PrimaryData	string	Data for PrimaryStatus. See "PrimaryStatus"
SecondaryStatus	string	Codified secondary status of the flight (see also "SecondaryInfo"). Possible values : "postponed_from" : current flight flight postponed from "SecondaryData" flight and date, "terminal_changed" : passenger terminal have changed. See "SecondaryData",

		<p>"checkin_changed" : checkin desks have changed. See "SecondaryData", "gate_changed" : boarding gates have changed. See "SecondaryData", "boarding_expected" : boarding expected to begin at "SecondaryData" time, "boarding_last_call" : boarding in progress, calling late passengers, "delivery_expected" : baggage delivery expected at "SecondaryData" time, "delivery_delayed" : baggage delivery delayed, "delivery_waiting" : baggage delivery waiting (late), "delivery_in_progress" : baggage delivery in progress, "delivery_completed" : baggage delivery completed, "bag_return" : baggage return after late departure flight cancellation</p>
PrimaryData	string	Airline code (see "airlines" resource). This is the airline code a passenger sees on its boarding pass
SecondaryData	string	Data for SecondaryStatus. See "SecondaryStatus"
FR	LocalizedFIDSDData	Localized FIDS information in french
EN	LocalizedFIDSDData	Localized FIDS information in english

2.2.7.4 LocalizedFIDSDData description

Localized FIDS information:

Name	Type	Description
PrimaryInfo	string	Main human readable info (see also "PrimaryStatus")
SecondaryInfo	string	Secondary human readable info (see also "SecondaryStatus")

2.2.7.5 TimesData description

Detailed time informations for a flight:

Name	Type	Description
DHC	date	Dernier Horaire Connue (DHC). Either SIBT, EIBT, QRE, H10, ALDT, AIBT

		(in sequence) for arrival and SOBT, TOBT, TSAT, AOBT, ATOT (in sequence) for departure. The current estimated or actual time depending on current flight situation
SIBT	date	Scheduled In-Block Time (SIBT). Only for arrival flight. The time that an aircraft is scheduled to arrive at its parking position. This is what the passenger sees on its boarding pass
EIBT	date	Estimated In-Block Time (EIBT). Only for arrival flight. The estimated time that an aircraft will arrive at its parking position (equivalent to Airline/Handler ETA – Estimated Time of Arrival)
QRE	date	QRE. Only for arrival flight. The ATC estimated time that an aircraft will arrive at its parking position (comes after EIBT)
ELDT	date	Estimated LanDing Time (ELDT). Only for arrival flight. The estimated time that an aircraft will touchdown on the runway
H10	date	H-10 minutes (H10). Only for arrival flight. The estimated time that an aircraft will arrive at its parking position (comes after QRE)
ALDT	date	Actual LanDing Time (ALDT). Only for arrival flight. Actual time of landing in Paris
AIBT	date	Actual In-Block Time (ALDT). Only for arrival flight. Actual time of arrival on stand
SOBT	date	Scheduled Off-Block Time (SOBT). Only for departure flight. The time that an aircraft is scheduled to depart from its parking position. This is what the passenger sees on its boarding pass
TOBT	date	Target Off-Block Time (TOBT). Only for departure flight. The time that an Aircraft Operator or Ground Handler estimates that an aircraft will be ready, all doors closed, boarding bridge removed, push back vehicle available and ready to start up / push

		back immediately upon reception of clearance
TSAT	date	Target Start Up Approval Time (TSAT). Only for departure flight. The time taking into account TOBT, CTOT and/or the traffic situation that an aircraft can expect startup / push back approval
CTOT	date	Calculated Take Off Time (CTOT). Only for departure flight. A time calculated and issued by the appropriate Central Management unit, as a result of tactical slot allocation, at which a flight is expected to become airborne
ASAT	date	Actual Start Up Approval Time (ASAT). Only for departure flight. Time that an aircraft receives its engine start up approval
AOBT	date	Actual Off-Block Time. Only for departure flight. Time the aircraft pushes back / vacates the parking position (equivalent to Airline/Handlers ATD – Actual Time of Departure)
ATOT	date	Actual Take Off Time (ATOT). Only for departure flight. The time that an aircraft takes off from the runway
IsDelayed	boolean	"true" if flight is known to be delayed

2.2.7.6 RouteData description

Flight routing informations. A flight contains at least one occurrence for destination or origin:

Name	Type	Description
AirportCode	string	IATA airport code (see "airports" resource)
Time	date	Scheduled time at this airport (local time at airport)

2.2.7.7 CheckinData description

Detailed passenger check-in informations (departure flight only):

Name	Type	Description
TimeLimit	string	Time limit of check-in. Check-in is closed after this time

Desks	string	One or more check-in desks allocated to flight. If more than one, this indicates a range (two desks separated by an hyphen)
DesksPoild	string	POI Id (Point Of Interest) associated to desks (if available). Please refer to POI Web API documentation for further information
Zones	string	One or more check-in zones allocated to flight. If more than one, this indicates a range (two zones separated by an hyphen). Note that a checkin zone contains many checkin desks (thus desk information is more precise than zone information).
ZonesPoild	string	POI Id associated to zones (if available). Please refer to POI Web API documentation for further information
FR	LocalizedCheckinData	Localized FIDS information in french
EN	LocalizedCheckinData	Localized FIDS information in english

2.2.7.8 LocalizedCheckinData description

Localized FIDS information:

Name	Type	Description
Info	string	Human readable info for checkin

2.2.7.9 BoardingData description

Detailed passenger boarding informations (departure flight only):

Name	Type	Description
Gates	string	One or more boarding gates allocated to flight. If more than one, this indicates a range (two gates separated by an hyphen)
GatesPoild	string	POI Id associated to gates (if available). Please refer to POI Web API documentation for further information
Rooms	string	One or more boarding gates

		allocated to flight. If more than one, this indicates a range (two gates separated by an hyphen)
RoomsPoild	string	POI Id associated to rooms (if available). Please refer to POI Web API documentation for further information
Status	string	Boarding status. Possible values : "waiting" : boarding has not started yet, "in progress" : boarding is in progress, "last call" : boarding is in progress, calling late passengers, "closed" : boarding is closed
FR	LocalizedBoardingData	Localized FIDS information in french
EN	LocalizedBoardingData	Localized FIDS information in english

2.2.7.10 LocalizedBoardingData description

Localized FIDS information:

Name	Type	Description
Info	string	Human readable info for boarding

2.2.7.11 ArrivalData description

Detailed passenger boarding informations (departure flight only):

Name	Type	Description
Zones	string	One or more arrival zones allocated to flight. If more than one, this indicates a range (two zones separated by an hyphen)
ZonesPoild	string	POI Id associated to zones (if available). Please refer to POI Web API documentation for further information
FR	LocalizedArrivalData	Localized FIDS information in french
EN	LocalizedArrivalData	Localized FIDS information in english

2.2.7.12 LocalizedArrivalData description

Localized FIDS information:

Name	Type	Description
Info	string	Human readable info for arrival

2.2.7.13 BaggageDeliveryDatadescription

Detailed baggage delivery informations (arrival flight only) :

Name	Type	Description
Belts	string	One or more delivery belts allocated to flight for baggage delivery. If more than one, this indicates a range (two belts separated by an hyphen)
BeltsPoild	string	POI Id associated to belts (if available). Please refer to POI Web API documentation for further information
Status	string	Baggage delivery status. Possible values : "foreseen" : delivery not yet stated. Foreseen to start at "Begin" time and finish at "End" time, "delayed" : delivery delayed until further notice, or delivery stopped after being in progress, "waiting" : delivery waiting. Foreseen to start at "Begin" time but this time have been exceeded, "in progress" : delivery in progress. Started at "Begin" time and foreseen to finish at "End" time, "completed" : delivery completed. Started at "Begin" time and completed at "End" time
Begin	date	Begin time of delivery. See "Status" for usage
End	date	Ending time of delivery. See "Status" for usage
FR	LocalizedDeliveryData	Localized FIDS information in french
EN	LocalizedDeliveryData	Localized FIDS information in english

2.2.7.14 LocalizedArrivalData description

Localized FIDS information:

Name	Type	Description
Info	string	Human readable info for baggage delivery

2.2.7.15 Request

You can use this API with HTTP GET.

	Request URI	Version
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey={apiKey}</i>	HTTP/1.1
GET	<i>http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey={apiKey}&flightDate={flightDate}&arrDep={arrDep}&terminalCode={terminalCode}&airlineCode={airlineCode}&destinationOrOriginAirportCode={destinationOrOriginAirportCode}&countryCode={countryCode}&flightNumber={flightNumber}&updatedSince={updatedSince}</i>	HTTP/1.1

You can use parameters

Parameters name	Type	Description
apiKey	string	API authentication token
id	string	Flight Ident
flightDate	date	Optional selector date of flight schedule (scheduled date of arrival at Paris or of departure from Paris)
arrDep	string	Optional selector for arrival only (A) or departure (D) (see ArrDep field)
terminalCode	string	Optional terminal code (see TerminalCode field)
airlineCode	string	Optional airline code (see AirlineCode field)
destinationOrOriginAirportCode	string	Optional airport code of origin or destination (see OriginAirportCode and DestinationAirportCode fields)
countryCode	string	Optional country code of origin or destination (see CountryCode field)
flightNumber	string	Optional flight number (only the number with an optional letter after, see FlightNumber field)
updatedSince	date	Optional time that will select only the flights for which the field "LastUpdate" is after the given time

2.2.7.16 Response

Response body:

```
{
  "ident": {
    "id": 12336644007,
    "airlineCode": "9W",
    "flightNumber": "6962"
  },
  "airportCode": "CDG",
  "arrDep": "D",
  "flightDate": "2015-04-11",
  "flightStatus": "active",
  "operating": {
    "id": 12218232397,
    "airlineCode": "AF",
    "flightNumber": "1260"
  },
  "originAirportCode": "CDG",
  "originFlightDate": "2015-04-11",
  "destinationAirportCode": "AMS",
  "destinationFlightDate": "2015-04-11",
  "serviceType": "J",
  "terminalCode": "C2F",
  "aircraftCode": "320",
  "registrationCode": "FGKXL",
  "runwayCode": "27L",
  "parkingCode": "F30",
  "busNeeded": false,
  "customs": "schengen",
  "lastUpdate": "2015-04-11T14:59:00",
  "fids": {
    "primaryStatus": "take_off",
    "primaryData": "14:58",
    "fr": {
      "primaryInfo": "Décollé à 14:58"
    },
    "en": {
      "primaryInfo": "Take off at 14:58"
    }
  },
  "times": {
    "dhc": "2015-04-11T14:58:00",
    "sobt": "2015-04-11T14:35:00",
    "tsat": "2015-04-11T14:49:00",
    "ctot": "2015-04-11T15:03:00",
    "aobt": "2015-04-11T14:46:00",
```



```
"atot": "2015-04-11T14:58:00",  
"isDelayed": false  
,  
"route": [  
  {  
    "airportCode": "AMS",  
    "time": "2015-04-11T15:55:00"  
  }  
,  
"boarding": {  
  "status": "closed"  
}  
}
```

3 Requests examples

All flight arrivals for today:

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&arrivalDate=2015-06-01>

All flight departures for today:

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&departureDate=2015-06-01>

All flight departures for a specific date (01/06/2015):

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&departureDate=2015-06-01>

All flight departures for a specific terminal (CDG 1):

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&terminalCode=C1>

All flight departures for a specific airline (AF = Air France):

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&airlineCode=AF>

Search for a specific flight (ex: AF084)

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&airlineCode=AF&flightNumber=084>

Search for a specific flight update after getting the flight from a list. Take the Id from "Flight/Ident/Id" and then pass this Id to API like this:

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights/12214656327?apiKey=yourKey>

Getting all possible destinations airports starting from Paris today :

- First get all the flights departing today :

<http://hackathonapi.aeroportsdeparis.fr/ADP.WebAPI.Flight/api/Flights?apiKey=yourKey&departureDate=2015-06-01>

- Then iterate thru all Route/RouteData and get all distinct "AirportCode". This will give you all destinations airports for today's flights.