json2puml

Make data visible and understandable

Jens Fudickar, April 2022

API’s and DATA

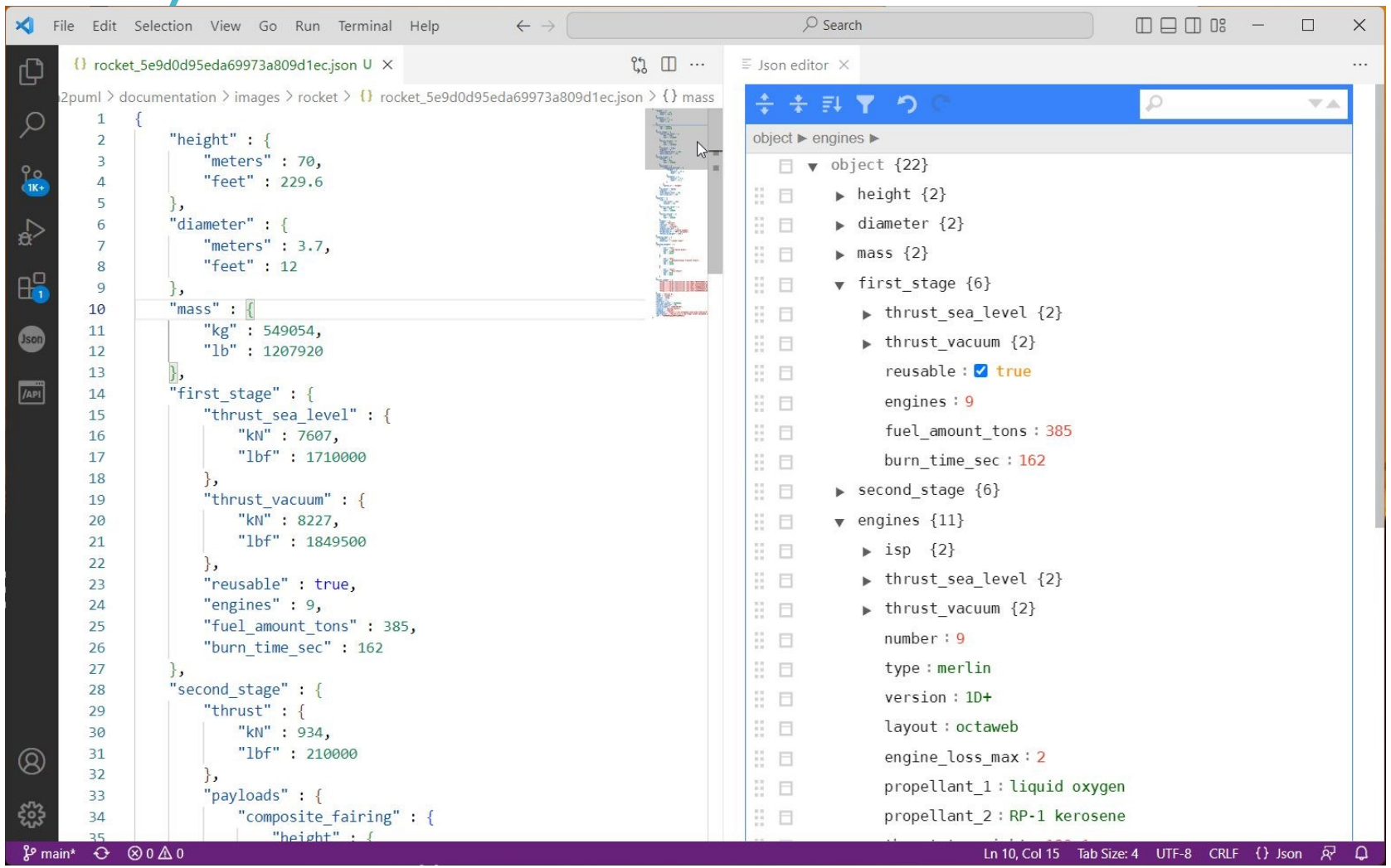
Working with API’s leads to working with JSON data

A typical answer of a REST API call can look like this:

{"height":{"meters":70,"feet":229.6},"diameter":{"meters":3.7,"feet":12},"mass":{"kg":549054,"lb":1207920},"first_stage":{"thrust_sea_level":{"kN":7607,"lbf":1710000},"thrust_vacuum":{"kN":8227,"lbf":1849500},"reusable":true,"engines":9,"fuel_amount_tons":385,"burn_time_sec":162},"second_stage":{"thrust":{"kN":934,"lbf":210000},"payloads":{"composite_fairing":{"height":{"meters":13.1,"feet":43},"diameter":{"meters":5.2,"feet":17.1}},"option_1":"dragon"},"reusable":false,"engines":1,"fuel_amount_tons":90,"burn_time_sec":397},"engines":{"isp":{"sea_level":288,"vacuum":312},"thrust_sea_level":{"kN":845,"lbf":190000},"thrust_vacuum":{"kN":914,"lbf":205500},"number":9,"type":"merlin","version":"1D+","layout":"octaweb","engine_loss_max":2,"propellant_1":"liquid oxygen","propellant_2":"RP-1 kerosene","thrust_to_weight":180.1},"landing_legs":{"number":4,"material":"carbon fiber"},"payload_weights":[{"id":"leo","name":"Low Earth Orbit","kg":22800,"lb":50265},{"id":"gto","name":"Geosynchronous Transfer Orbit","kg":8300,"lb":18300},{"id":"mars","name":"Mars Orbit","kg":4020,"lb":8860}],"flickr_images":["https:\/\/farm1.staticflickr.com\/929\/28787338307_3453a11a77_b.jpg","https:\/\/farm4.staticflickr.com\/3955\/32915197674_eee74d81bb_b.jpg","https:\/\/farm1.staticflickr.com\/293\/32312415025_6841e30bf1_b.jpg","https:\/\/farm1.staticflickr.com\/623\/23660653516_5b6cb301d1_b.jpg","https:\/\/farm6.staticflickr.com\/5518\/31579784413_d853331601_b.jpg","https:\/\/farm1.staticflickr.com\/745\/32394687645_a9c54a34ef_b.jpg"],"name":"Falcon 9","type":"rocket","active":true,"stages":2,"boosters":0,"cost_per_launch":50000000,"success_rate_pct":98,"first_flight":"2010-06-04","country":"United States","company":"SpaceX","wikipedia":"https:\/\/en.wikipedia.org\/wiki\/Falcon_9","description":"Falcon 9 is a two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of satellites and the Dragon spacecraft into orbit.","id":"5e9d0d95eda69973a809d1ec"}

How to visualize / understand this ?

(Online) JSON Editor

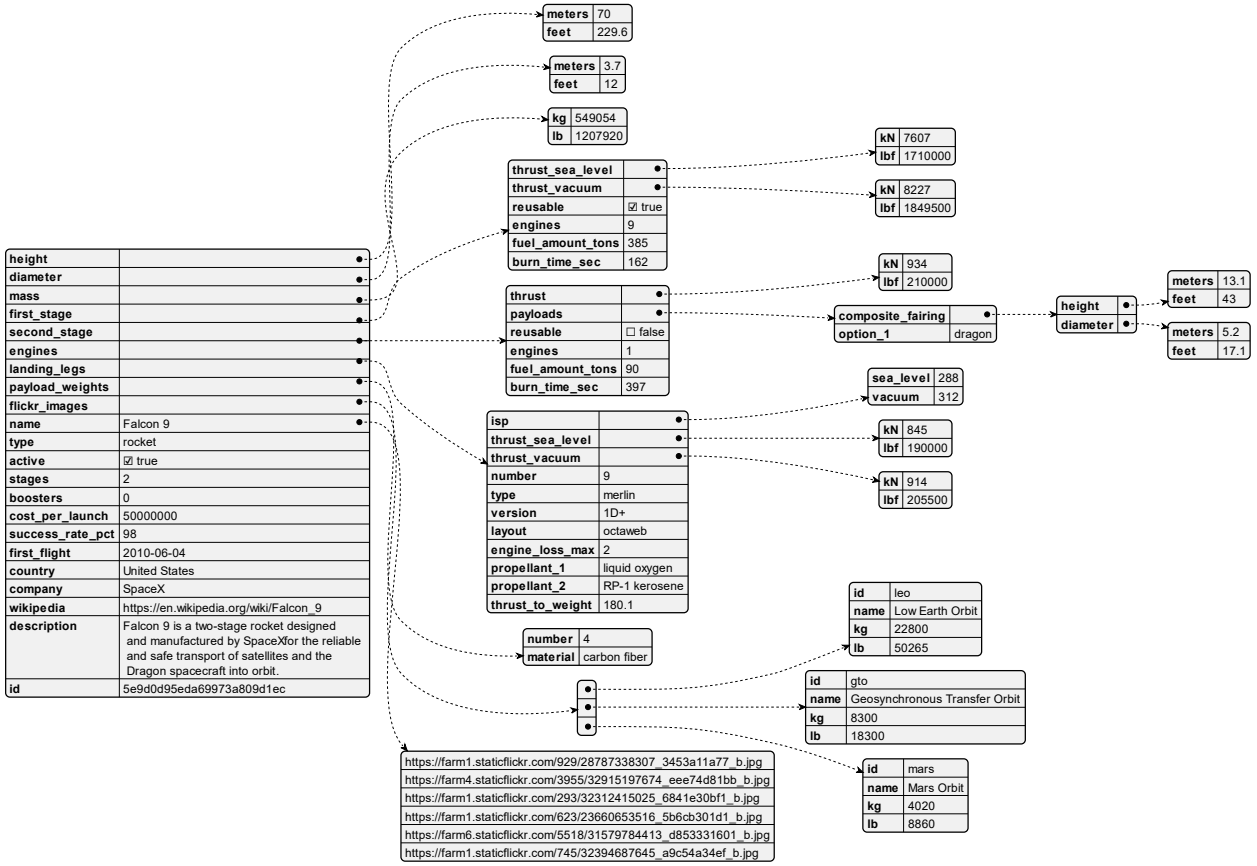


This makes the live easier but it is still complex ☺

Confluence and Standard PlantUML

PlantUML supports an OOTB visualisation of JSON.

```
@startjson
<json>
@endjson
```

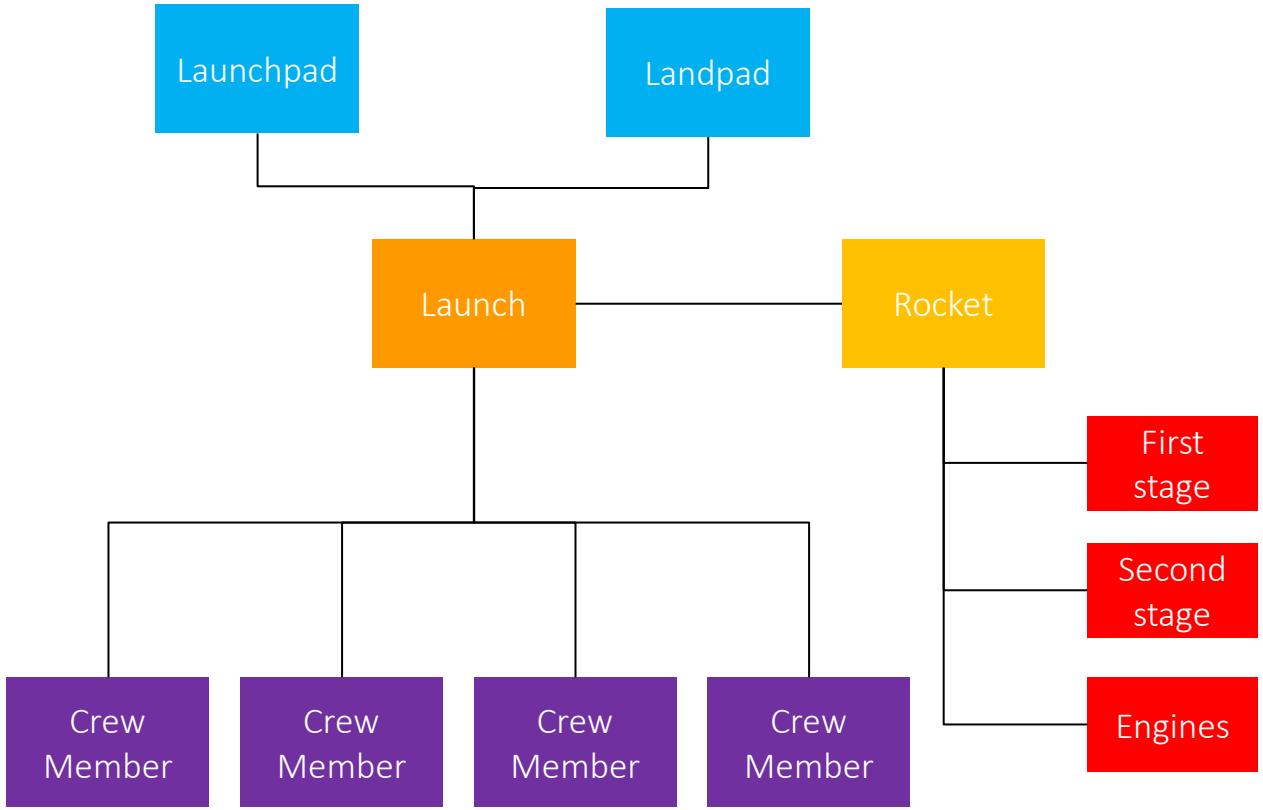


This improves also, but there is no knowledge about the data models behind

Drawing by Hand

It’s nice, and it’s giving you a task.

But it’s not effective, not accurate and not fast enough.



One Solution: `json2puml`

`Json2puml` is a command line tool developed to generate PlantUML files based JSON files.

`Json2puml` has an understanding of how data is structured and simplifies and visualises the outcome.

`json2puml` has the possibility to combine the JSON results of multiple API calls into one result set.

`json2puml` is highly configurable to generate outcomes in different detailed levels.

`json2puml` is free to use for everyone.

SpaceX REST API

Open Source REST API for launch, rocket, core, capsule, starlink, launchpad, and landing pad data.

https://github.com/r-spacex/SpaceX-API

What is this?

It’s an unofficial public API to receive various information about all Space X rocket launches.

Space X API – get /rocket

One API result formatted with three different options.

rocket\rocket_5e9d0d95eda69973a809d1ec.compact.puml

Ⓡ rocket **Falcon 9**

rocket - 5e9d0d95eda69973a809d1ec

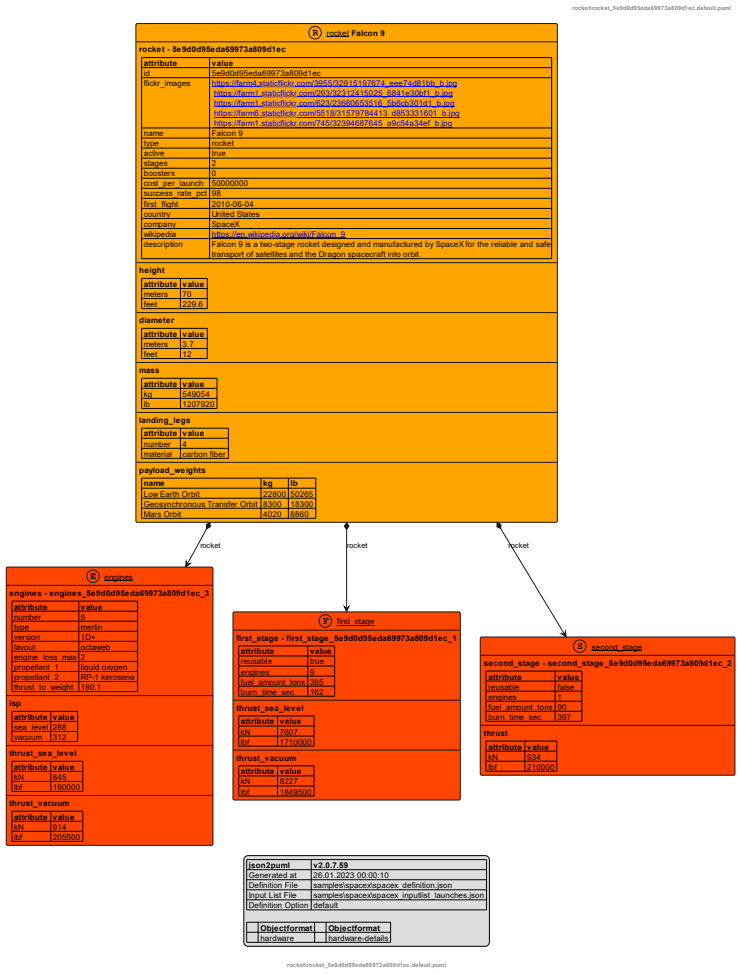
attribute	value
id	5e9d0d95eda69973a809d1ec
name	Falcon 9
type	rocket
active	true
stages	2
boosters	0
cost_per_launch	50000000
success_rate_pct	98
first_flight	2010-06-04
country	United States
company	SpaceX

json2puml	**v2.0.7.59**
Generated at	26.01.2023 00:50:08
Definition File	samples\spacex\spacex_definition.json
Input List File	samples\spacex\spacex_inputlist_launches.json
Definition Option	compact

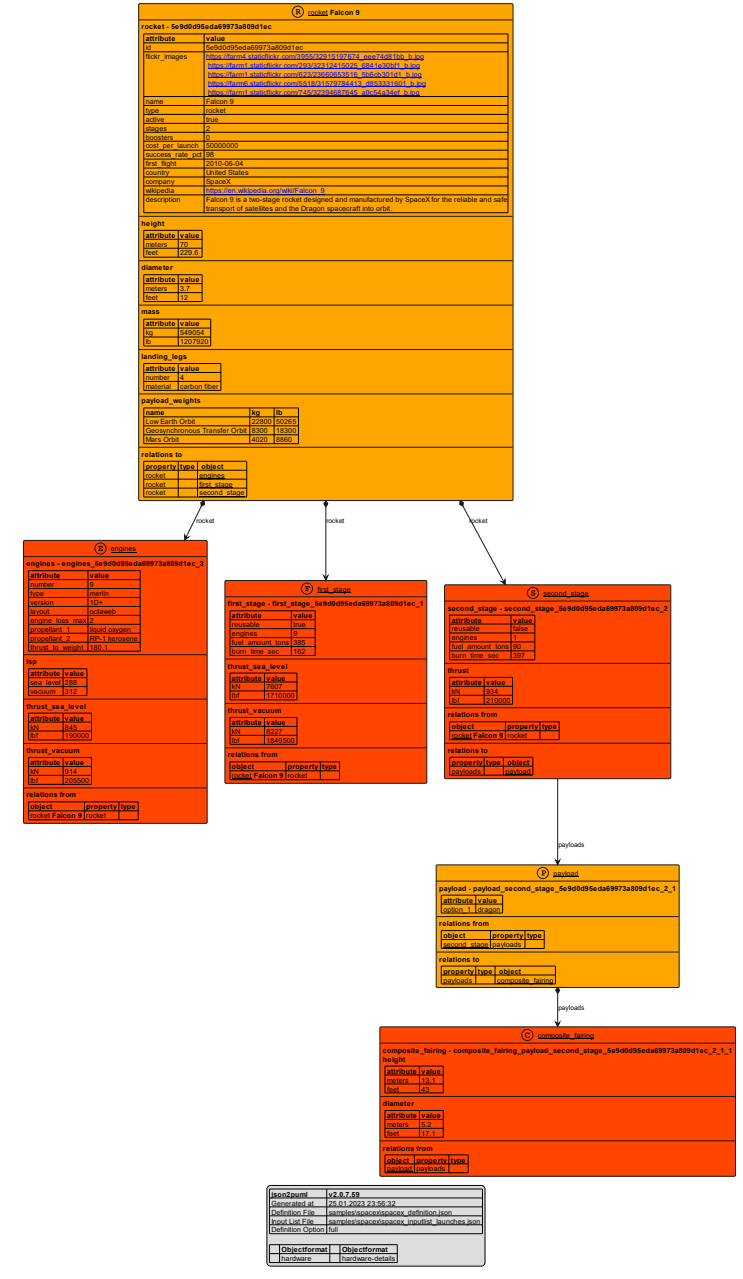
	Objectformat
	hardware

rocket\rocket_5e9d0d95eda69973a809d1ec.compact.puml

Option : compact

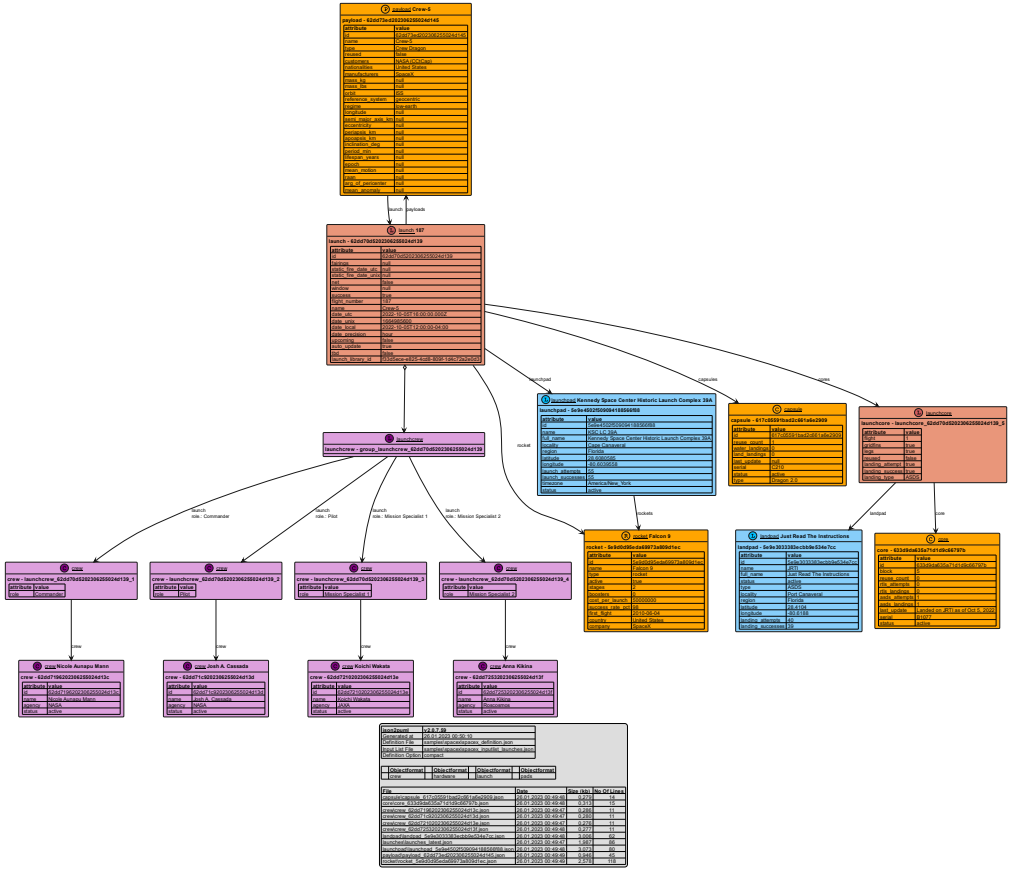


Option : default

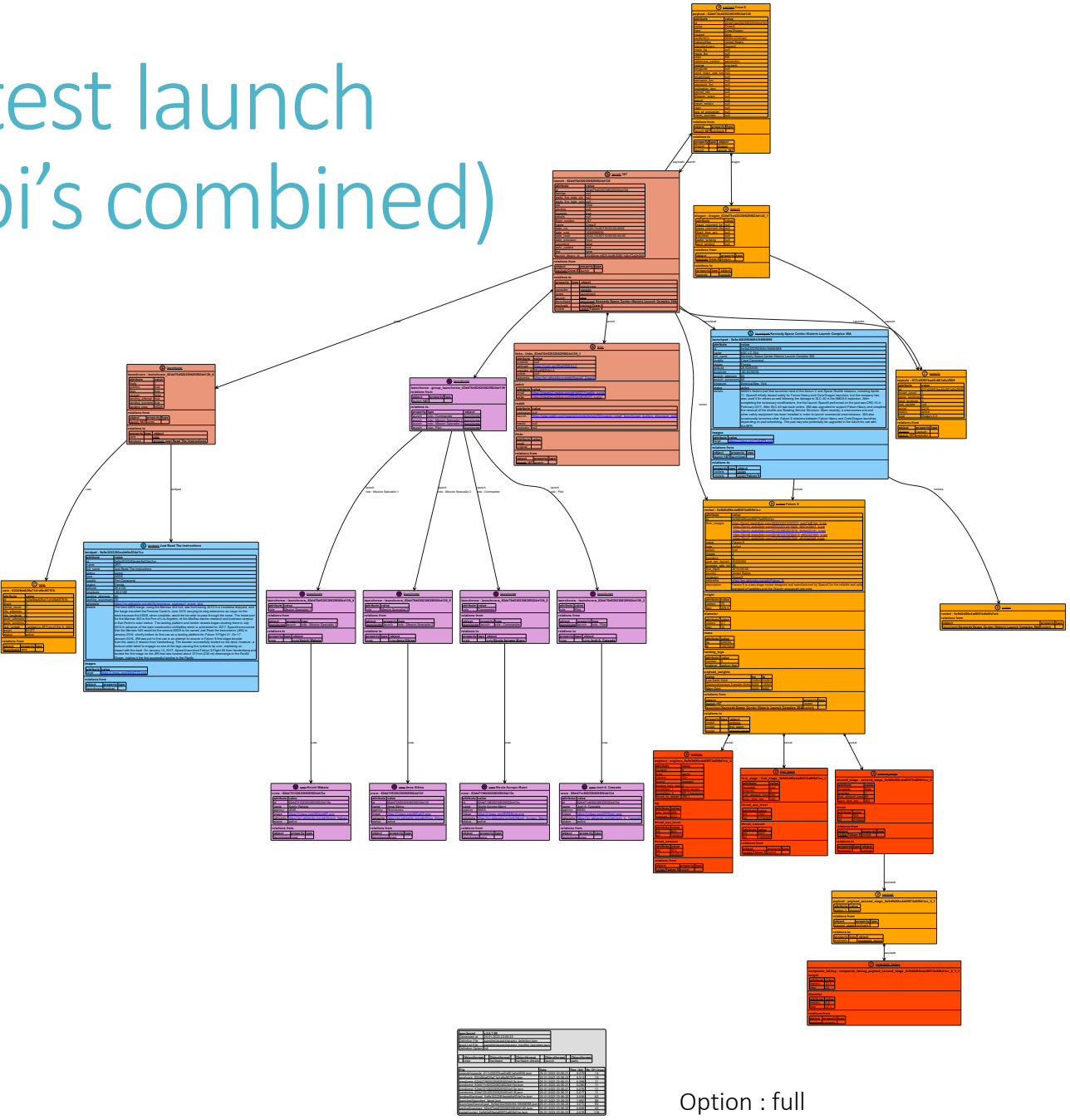


Option : full

All data regarding the latest launch of a Space X rocket (8 Api's combined)



Option : compact



Option : full

SWAPI - The Star Wars API

https://swapi.dev/

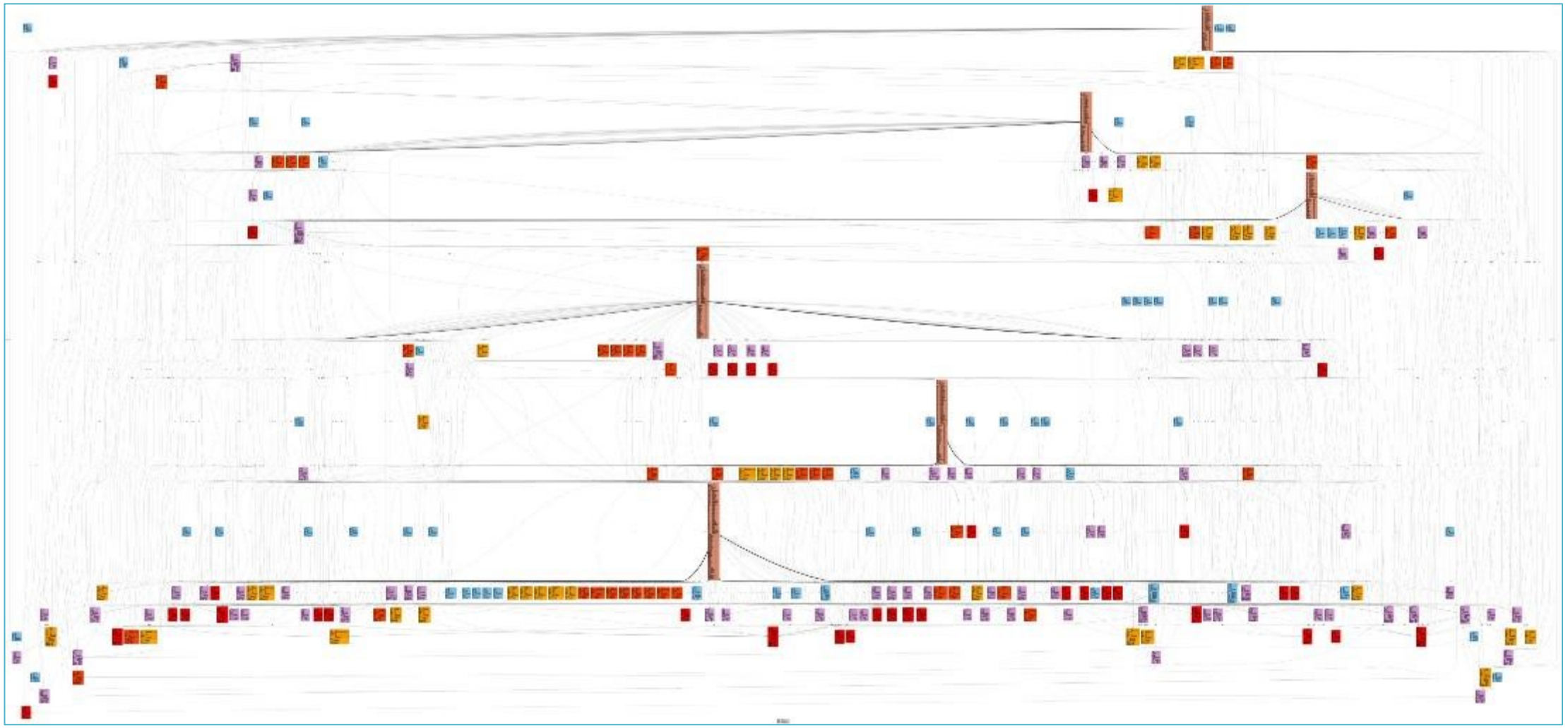
What is this?

The Star Wars API, or "swapi" (Swah-pee) is the world's first quantified and programmatically-accessible data source for all the data from the Star Wars canon universe!

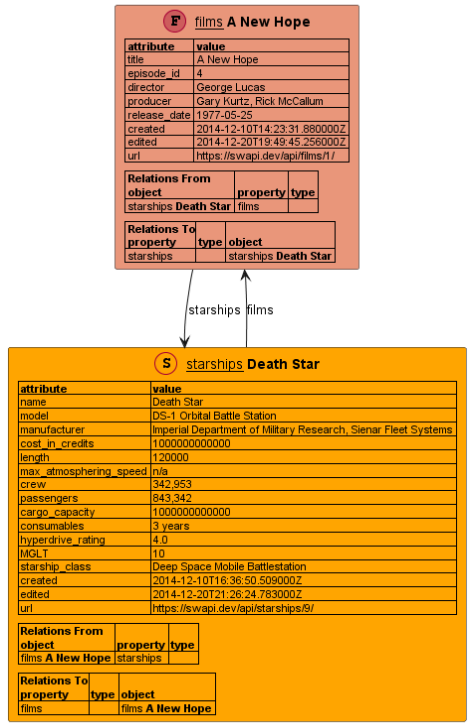
We've taken all the rich contextual stuff from the universe and formatted into something easier to consume with software. Then we went and stuck an API on the front so you can access it all!

All Data

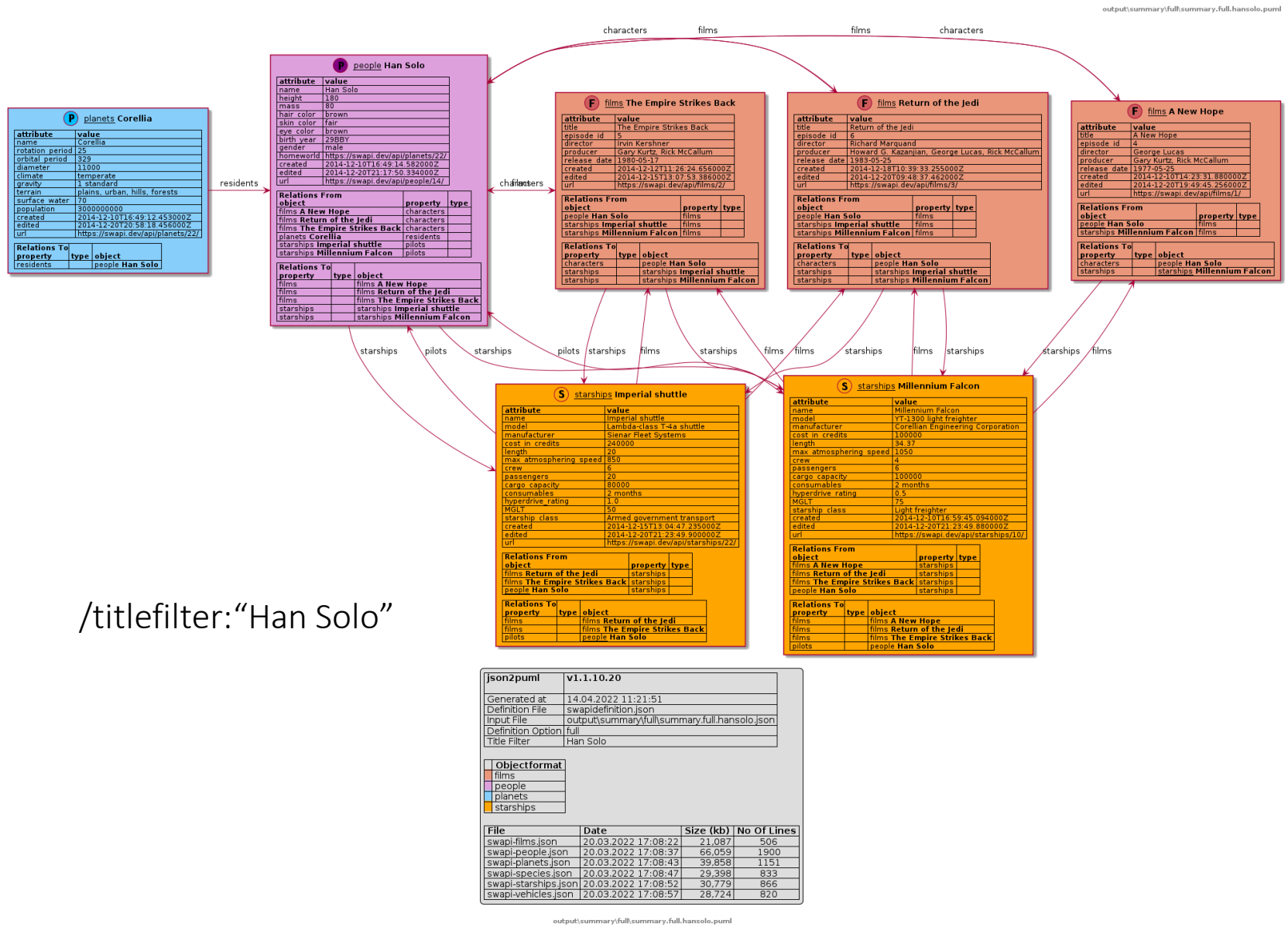
6 Films, 82 Persons, 60 Planets, 37 Species, 36 Star ships, 39 Vehicles



Examples



/titlefilter:“Death Star”



/titlefilter:“Han Solo”

C2 General

How is it working

`json2puml` Has three main components which can be combined

1. Fetch data from a defined set of API's using curl. (Optional)
2. Convert the JSON data into a PlantUml script.
3. Convert the PlantUml script into a SVG or PNG file using PlantUml command line utility.

`json2puml` can automate the fetching of data by using data from an API result as an input criteria for the next API call.
E.g. Using the SpaceX API the `get /launch` result includes the ID of the used rocket. This ID can be used for the next `get /rocket` call.

Operating Systems

`json2puml` can be used on MS Windows and on Linux.

For Linux there are plain executables and docker container available.

`json2puml` can run as command line tool or as a service application answering on http based REST calls.

Where can you find it:

It’s published on Github.com under a GPLv3 license.

https://github.com/jfudickar/json2puml