

Exercise 6

Distributed Software Systems — Prof. Paolo Ciancarini
Università di Bologna
A.Y. 2023/2024
Lorenzo Campidelli, Riccardo Scotti

Mashup

We split the work between us, one worked on the webapp and the other on the mashup.

To fetch data for the mashup we used the service from weatherapi.com for current weather and the next 24 hours, and because the site asks for a paid subscription for daily forecast the data for the next 5 days are fetched via the webapp api.

Of course different sources deliver different data, so what can be displayed will depend on the source of the data.

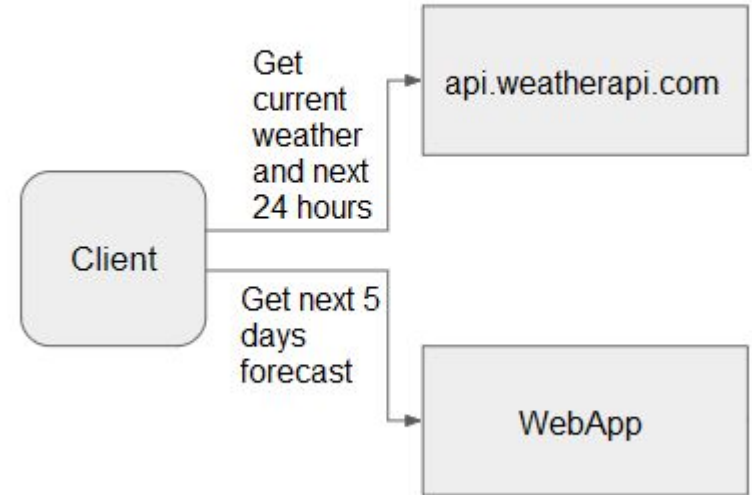
The mashup is made in html, using javascript and jquery for page organization and ajax calls to the apis, and css for the style.

Mashup: architecture

The architecture is very simple, the client requests data from two different sources and displays them in an html file.

This model can be easily expanded with data from additional sources, which can be combined with existing ones.

The data used is relative only to the city of Bologna, but both apis used can manage every requested location.



Mashup: current weather

Weather in Bologna

Current weather

Light rain



Temperature: 10°C

Humidity: 94%

Precipitations: 2.69mm

























Wind: 6.8kph

Wind direction: W

Last updated: 2023-11-21 18:30

Mashup: next 24 hours

Next 24 hours

| | | | | | | | |
|---|---|--|--|--|---|---|---|
| <p>Hour: 19:00</p>  <p>Light drizzle</p> <p>Temperature: 9.6°C</p> <p>Rain chance: 96%</p> | <p>Hour: 20:00</p>  <p>Light rain</p> <p>Temperature: 9.4°C</p> <p>Rain chance: 96%</p> | <p>Hour: 21:00</p>  <p>Patchy rain possible</p> <p>Temperature: 9.3°C</p> <p>Rain chance: 95%</p> | <p>Hour: 22:00</p>  <p>Patchy rain possible</p> <p>Temperature: 9.2°C</p> <p>Rain chance: 67%</p> | <p>Hour: 23:00</p>  <p>Patchy rain possible</p> <p>Temperature: 8.8°C</p> <p>Rain chance: 74%</p> | <p>Hour: 0:00</p>  <p>Patchy rain possible</p> <p>Temperature: 8.5°C</p> <p>Rain chance: 84%</p> | <p>Hour: 1:00</p>  <p>Patchy rain possible</p> <p>Temperature: 8.4°C</p> <p>Rain chance: 78%</p> | <p>Hour: 2:00</p>  <p>Cloudy</p> <p>Temperature: 8.3°C</p> <p>Rain chance: 0%</p> |
| <p>Hour: 3:00</p>  <p>Overcast</p> <p>Temperature: 9°C</p> <p>Rain chance: 0%</p> | <p>Hour: 4:00</p>  <p>Partly cloudy</p> <p>Temperature: 9.3°C</p> <p>Rain chance: 0%</p> | <p>Hour: 5:00</p>  <p>Overcast</p> <p>Temperature: 8.8°C</p> <p>Rain chance: 0%</p> | <p>Hour: 6:00</p>  <p>Overcast</p> <p>Temperature: 8.7°C</p> <p>Rain chance: 0%</p> | <p>Hour: 7:00</p>  <p>Patchy rain possible</p> <p>Temperature: 8.6°C</p> <p>Rain chance: 76%</p> | <p>Hour: 8:00</p>  <p>Patchy rain possible</p> <p>Temperature: 8.7°C</p> <p>Rain chance: 87%</p> | <p>Hour: 9:00</p>  <p>Patchy rain possible</p> <p>Temperature: 9.4°C</p> <p>Rain chance: 68%</p> | <p>Hour: 10:00</p>  <p>Patchy rain possible</p> <p>Temperature: 11.1°C</p> <p>Rain chance: 79%</p> |
| <p>Hour: 11:00</p>  <p>Patchy rain possible</p> <p>Temperature: 12°C</p> <p>Rain chance: 75%</p> | <p>Hour: 12:00</p>  <p>Patchy rain possible</p> <p>Temperature: 12.4°C</p> <p>Rain chance: 89%</p> | <p>Hour: 13:00</p>  <p>Sunny</p> <p>Temperature: 12.8°C</p> <p>Rain chance: 0%</p> | <p>Hour: 14:00</p>  <p>Patchy rain possible</p> <p>Temperature: 13°C</p> <p>Rain chance: 62%</p> | <p>Hour: 15:00</p>  <p>Sunny</p> <p>Temperature: 12.5°C</p> <p>Rain chance: 0%</p> | <p>Hour: 16:00</p>  <p>Sunny</p> <p>Temperature: 11.3°C</p> <p>Rain chance: 0%</p> | <p>Hour: 17:00</p>  <p>Clear</p> <p>Temperature: 9.8°C</p> <p>Rain chance: 0%</p> | <p>Hour: 18:00</p>  <p>Clear</p> <p>Temperature: 9.2°C</p> <p>Rain chance: 0%</p> |

Mashup: next 5 days

Next 5 days

Day: 2023-11-22
Temperatures:
Min: 6.5
Max: 12.5
Sunrise: 07:19
Sunset: 16:41

Day: 2023-11-23
Temperatures:
Min: 2.7
Max: 10.6
Sunrise: 07:21
Sunset: 16:41

Day: 2023-11-24
Temperatures:
Min: 1.8
Max: 11.8
Sunrise: 07:22
Sunset: 16:40

Day: 2023-11-25
Temperatures:
Min: 4.4
Max: 10.5
Sunrise: 07:23
Sunset: 16:39

Day: 2023-11-26
Temperatures:
Min: -0.7
Max: 9.4
Sunrise: 07:24
Sunset: 16:39

Data provided from this api does not have condition icons, but it has sunrise and sunset informations and min and max temperatures for the day.

All these data can be combined with the previous ones (with proper synchronization between the two different http calls), but data from weatherapi is limited in number of days so this section cannot use data from it.

Web service

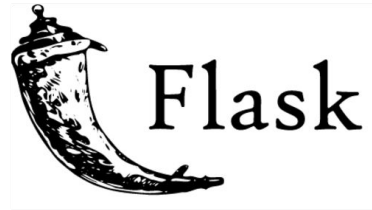
We have developed a web service to provide weather data through an API, accessible via HTTP requests.

The web service takes its data from OpenMeteo, an open source API which offers not only **current data**, but also **forecast**, directly from reliable sources such as Deutscher Wetterdienst, US weather service (NOAA) and Meteo France, among others.

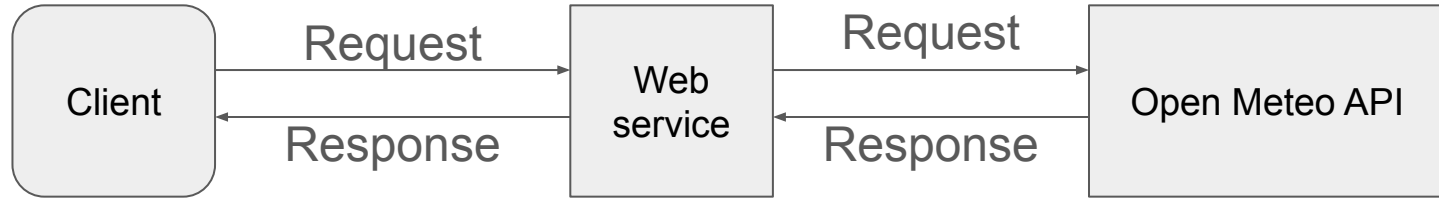


Web service: implementation

- The web service is entirely written in Python.
- **Flask** is used to handle HTTP requests.
- Instead of directly querying the Open Meteo API with HTTP requests, it uses the open source wrapper package [open-meteo](#).
- Additionally, it uses [Flask-RESTX](#) to facilitate RESTful API development and documentation.



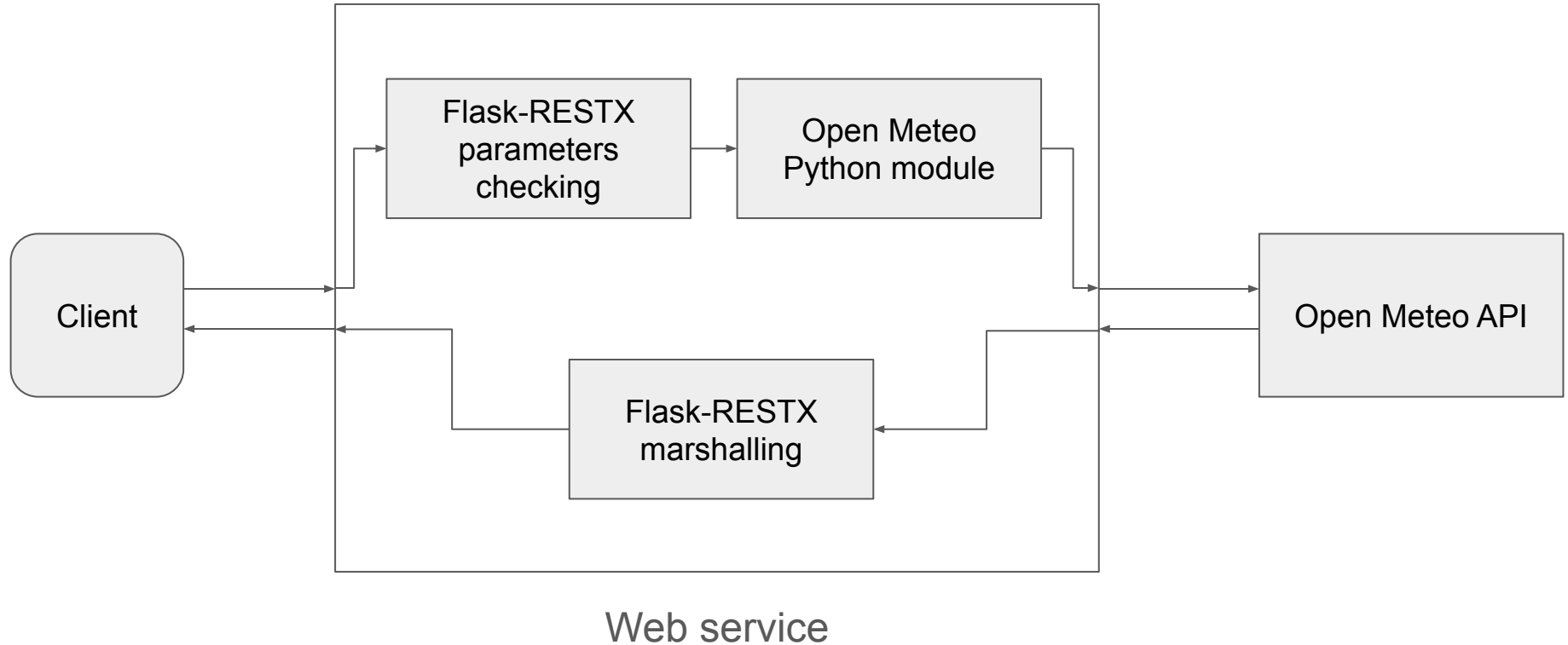
Web service: architecture



Viewing the web service from the outside, as a black box, it may seem it is merely an interface for Open Meteo API and it simply enriches and forwards incoming requests.

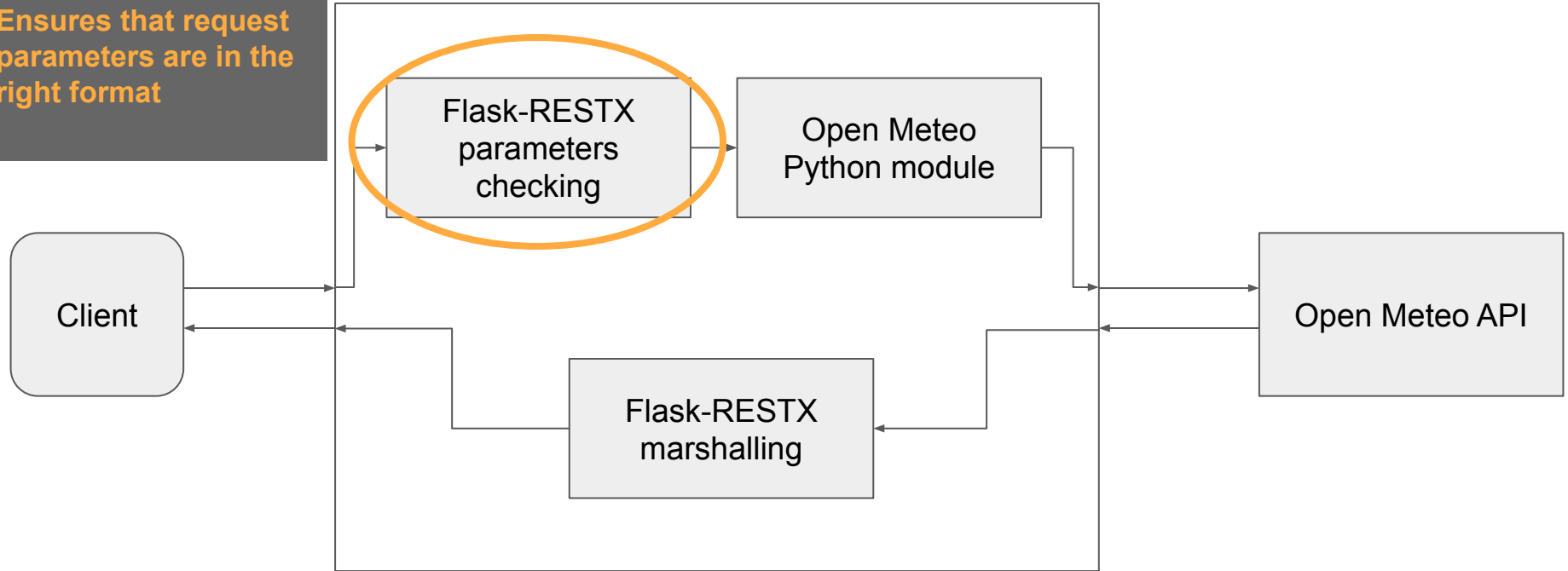
In reality, if we look at the modules that compose the service, we can see that it offers additional functionalities, regarding the simplicity of its usage, the error handling and the documentation.

Web service: architecture



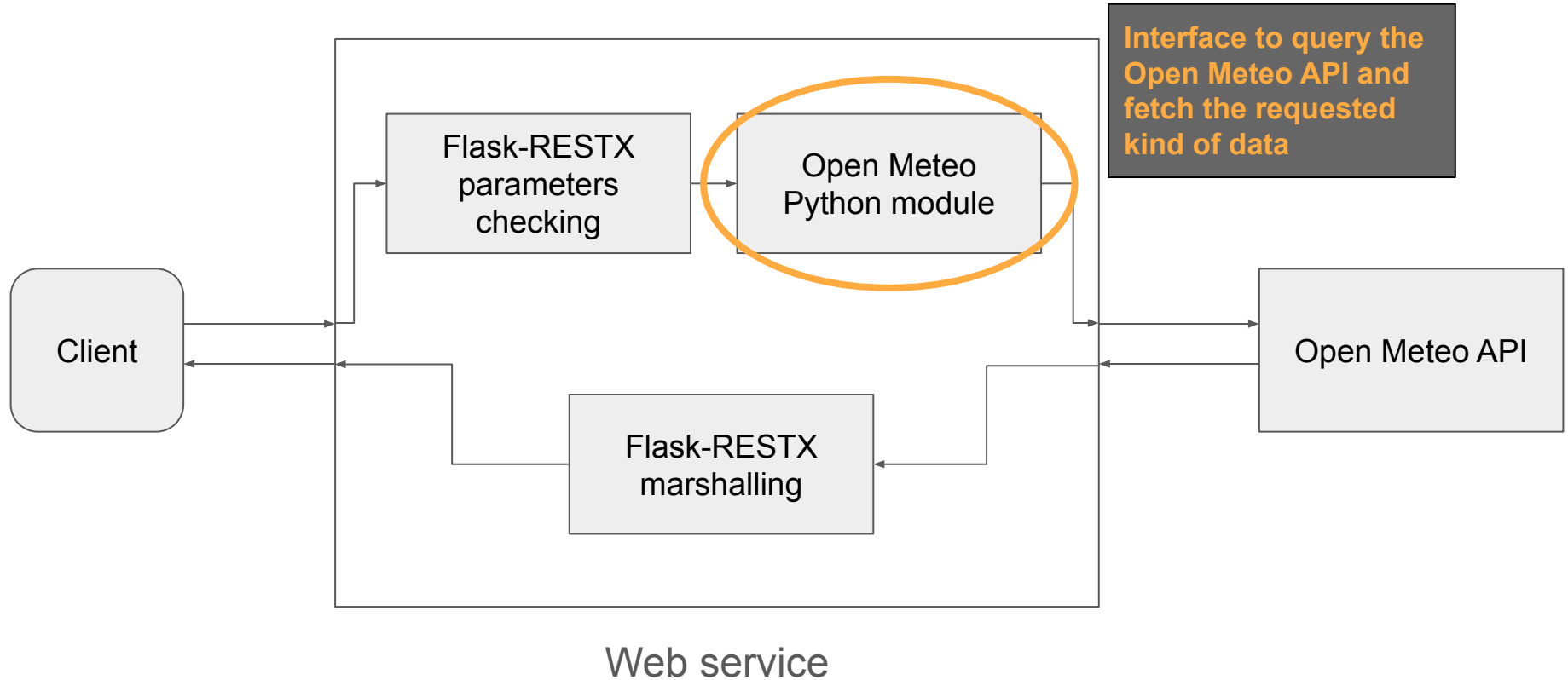
Web service: architecture

Ensures that request parameters are in the right format

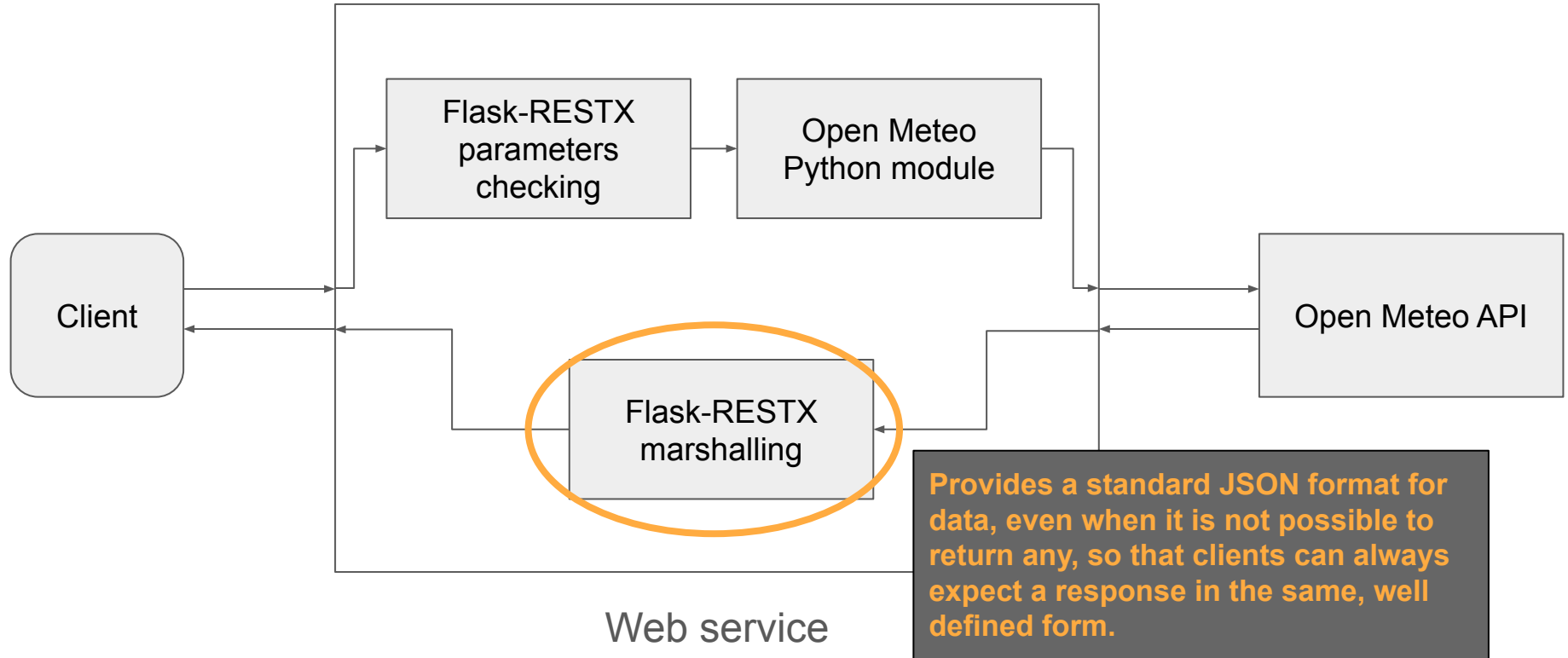


Web service

Web service: architecture



Web service: architecture



Web service: API endpoints

- POST instead of GET, to simplify sending parameters
- Simple usage, only requires location and possibly number of days of forecast
- Documentation automatically generated by Flask-RESTX, available as a webpage

POST

/api/current Returns current weather data for a specific location

POST

/api/daily Returns weather data hour by hour in a given day

POST

/api/hourly Returns weather data hour by hour in a given day

Web service: API documentation and testing

| Models | |
|--------------------------|--------|
| Location ▾ { | |
| lat | number |
| lon | number |
| } | |
| Weather data ▾ { | |
| datetime | string |
| minTemperature | number |
| maxTemperature | number |
| temperature | number |
| windSpeed | number |
| windDirection | number |
| humidity | number |
| precipitation | number |
| } | |
| Hourly forecast > | |
| Hour data model > | |
| Daily forecast request > | |
| Daily forecast > | |

The automatically generated documentation includes not only the endpoints, but also models for data to be sent and received.

This makes working with the application very easy and clear for programmers that want to use the service, and offer a uniform data marshalling format, making it possible to exploit the API regardless of the language or technology used in the client.

Web service: documentation and testing

The documentation web page generated by Flask-RESTX serves also as a platform to easily test the API endpoints (similarly to applications like Postman/Insomnia).

POST **/api/current** Returns current weather data for a specific location

Parameters

| Name | Description |
|--|---|
| payload <small>★ required</small> object (body) | Edit Value Model <pre>{ "lat": 44.4938203, "lon": 11.3426327 }</pre> |

Coordinates for Bologna

Server response

| Code | Details |
|------|--|
| 200 | <div>Response body<pre>{ "datetime": "2023-11-22T13:00:04.392266", "minTemperature": 7.3, "maxTemperature": 14.6, "temperature": 14.6, "windSpeed": 21.1, "windDirection": 57, "humidity": 45, "precipitation": 0 }</pre></div> <div>Response headers<pre>connection: close content-length: 227 content-type: application/json date: Wed, 22 Nov 2023 12:00:04 GMT server: Werkzeug/2.3.8 Python/3.10.12</pre></div> |

Current weather data

Web service: documentation and testing

The documentation web page generated by Flask-RESTX serves also as a platform to easily test the API endpoints (similarly to applications like Postman/Insomnia).

| Name | Description |
|--|--|
| payload <small>* required</small> | |
| object | Edit Value Model |
| (body) | |
| | <pre>{ "numDays": 2, "loc": { "lat": 40.734205, "lon": 13.947125 } }</pre> |

Coordinates for Ischia

| Server response | |
|---|---------------|
| Code | Details |
| 200 | Response body |
| <pre>[{ "date": "2023-11-22", "minTemperature": 15.1, "maxTemperature": 17.2, "sunrise": "06:58:00", "sunset": "16:41:00", "precipitation": "[4.4, 0.0, 1.3, 16.6, 0.0, 0.0, 0.0]" }, { "date": "2023-11-23", "minTemperature": 16, "maxTemperature": 17.9, "sunrise": "07:00:00", "sunset": "16:41:00", "precipitation": "[4.4, 0.0, 1.3, 16.6, 0.0, 0.0, 0.0]" }]</pre> | |

Weather forecast for the next 2 days

Web service: documentation and testing

The documentation web page generated by Flask-RESTX serves also as a platform to easily test the API endpoints (similarly to applications like Postman/Insomnia).

| Name | Description |
|--|--|
| payload ★ required | |
| object (body) | <div>Edit Value Model</div> <pre>{ "lat": 47.6038321, "lon": -122.330062 }</pre> |

Coordinates for Seattle (USA)

| Code | Details |
|------|--|
| 200 | <div>Response body</div> <pre>{ "date": "2023-11-22T13:07:23.708674", "minTemperature": 7.3, "maxTemperature": 10.5, "hours": [{ "temperature": 10.5, "windSpeed": 11.2, "windDirection": 159, "humidity": 60, "precipitation": 0 }, { "temperature": 10.4, "windSpeed": 10.5, "windDirection": 172, "humidity": 58, "precipitation": 0 }] }</pre> |

Hourly forecast for the current day

Weather mashup and web service – comparison

| | Weather mashup | Weather web service |
|-------------------------|--|---|
| Ease of use | User friendly, data is shown automatically (in our case even without any input) | Unusable for the final user, but our service is easier for programmers than directly using more complex APIs (such as Open Meteo) |
| Reusability | Practically non-existent | High, designed specifically for it |
| Control over shown data | Depends on what the developer decides to show (ex. forecast for a certain area only as in this case) | In general, more freedom (at the cost of more parameters to send and more data to look through) |