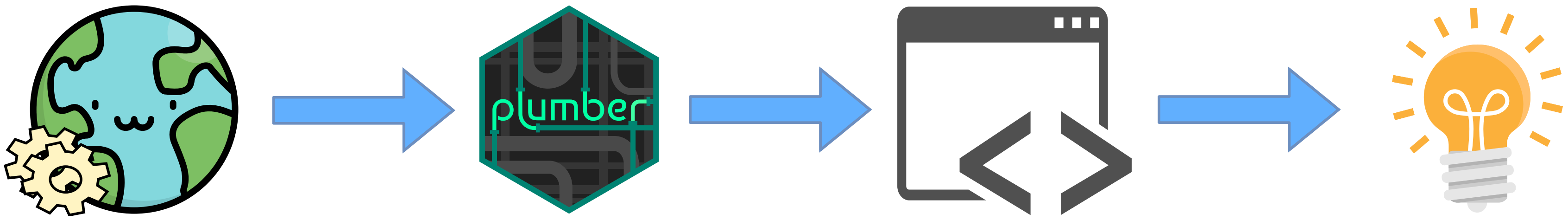
A Web Service for Processing Location Data from Mobile Phones

Stefan Fleck

2020-12-04

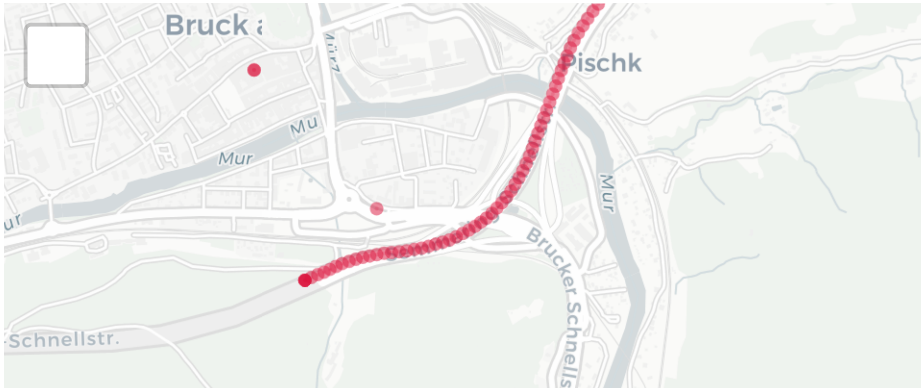
Outline

1. Processing location data
2. Web services with R
3. Architecture and code examples
4. Conclusion

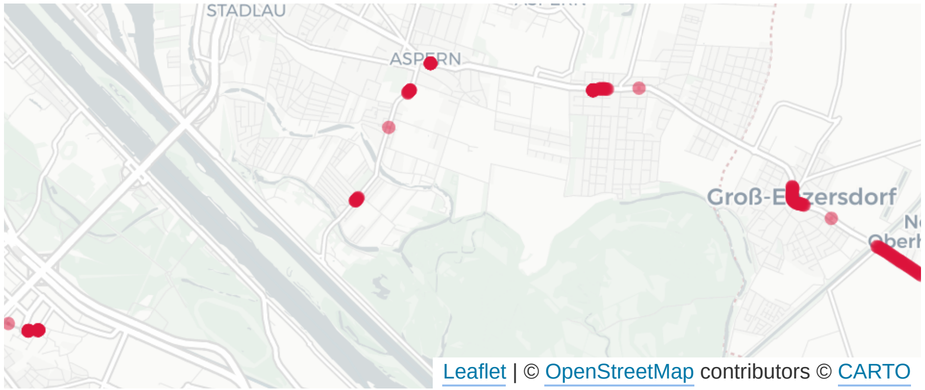


Travel distance from position data

- The mobile phone records:
  - Geo-position every *x* seconds (varies by device)
  - Positional accuracy
  - Speed
- Outlier-cleanup necessary
- Imputation of gaps necessary (via routing)



Outliers



Leaflet | © OpenStreetMap contributors © CARTO

Gaps

Travel distance from position data II

1. Remove points where
   - position accuracy is less than 20m
   - reported speed is less than 7kph (walking)
   - calculated speed is greater than 130kph (outliers)
2. Connect points that are close together with straight lines
3. Use external routing service to connect points further than 50m apart

reported speed: Speed according to speed sensor of the device

calculated speed: Speed calculated from position & timestamp

How do we integrate our R code in production?

Problem: A new journey can be submitted at any time of the day and needs to be processed within minutes

- 🐒 Manually execute the script each time we receive a new journey
- 👨 Write an R-Script that is callable from the the command line
  - Makes it hard to ensure a stable R environment
  - Maintenance of the script involves another party
- 🚀 Turn your R-code into a web-service!

Advantages of web services

- Clear separation of concerns (“good fences make good neighbors”)
- Virtually all programming languages can use HTTP APIs
- Execution environment can be tightly controlled
- Easy to deploy different version of the service (dev, test, prod)
- A service is reusable and not limited to a single project
- (for our project) We could re-use existing code from a different project

Web Service: A service that runs on a server and responds to requests (usually via HTTP)

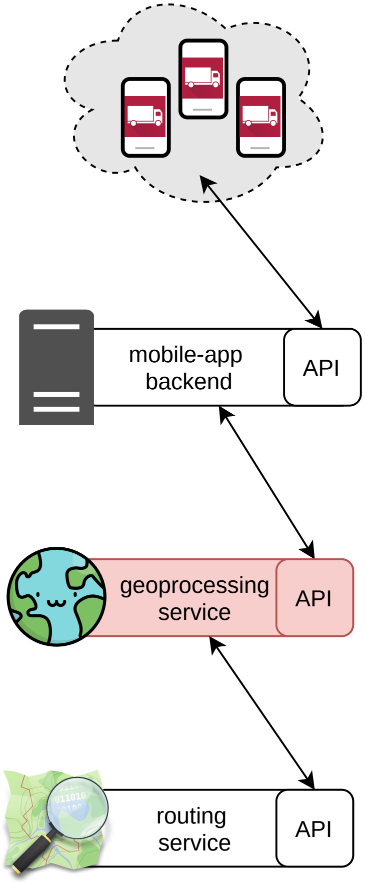
HTTP: A protocol for computers to communicate with each other

API: Application Programming Interface: A part of a program that enables it to communicate with other programs

Architecture

- Mobile-App communicates with backend
- Backend communicates with geoprocessing service
- Geoprocessing service communicates with external routing service

Each component can easily be replaced as long as its API stays stable!



Server code

```r
#* Calculate the fastest route
#* @param ride_id integer id
#* @response 400 Impossible Route Rrror: routing not possible (e.g. Islands)
#* @response 504 Gateway Timeout Error: routing backend may be down.
#*
#* @get /v1/rides/<id>/summary
rides_summary <- function(
  id = "",  # from the path definition above
  res   # special plumber object: the Response
){
  r <- tryCatch({
    calculate_fastest_route(id)
  },
    impossible_route_error = function(e){res$status <- 400},
    gateway_timeout_error = function(e){res$status <- 504}
  )

  r
}
```

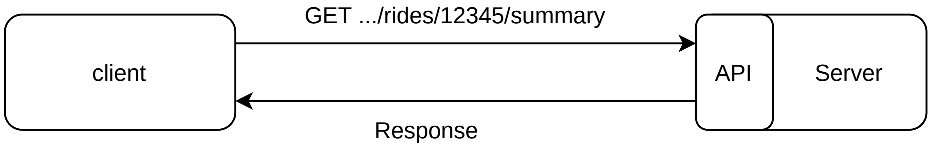
Client code

Sending a GET request from R

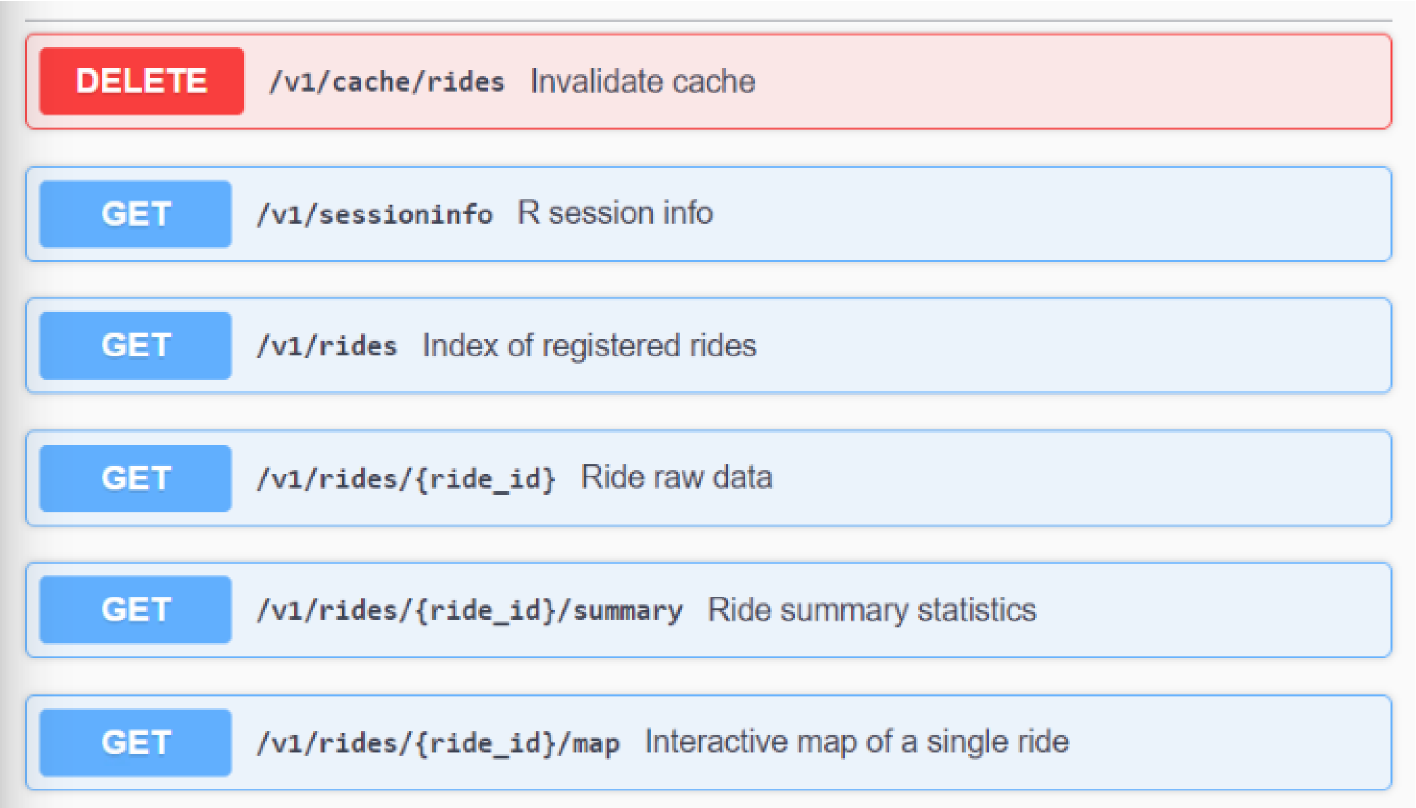
```
res <- httr::GET("https://<...>/rides/12345/summary")
httr::content(res)

## $distanceAustria
## [1] 297334.3
##
## $distanceForeign
## [1] 0
```

Or from the command line

```
curl https://<...>/rides/12345/summary
```



API Documentation



DELETE `/v1/cache/rides` Invalidate cache

GET `/v1/sessioninfo` R session info

GET `/v1/rides` Index of registered rides

GET `/v1/rides/{ride_id}` Ride raw data

GET `/v1/rides/{ride_id}/summary` Ride summary statistics

GET `/v1/rides/{ride_id}/map` Interactive map of a single ride

HTTP request methods

OpenAPI specification

Conclusion

- HTTP APIs are a powerful tool for patching applications together
- Providing services with well-behaved APIs is a great way for different teams to collaborate on complex IT projects
- The **plumber** package makes it easy to create HTTP APIs for your project
- The **httr** package makes it easy to access HTTP APIs
- Use APIs!

> *"Anyone who doesn't do this will be fired. Thank you; have a nice day!"*
>
> – Jeff Bezos

Further reading

plumber: An API Generator for R

httr: Tools for Working with URLs and HTTP

HTTP: The Protocol Every Web Developer Must Know

Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

What is REST

The Austrian Road Freight Transport Mobile App - Video or Brochure [in German]

This Presentation

Contact

stefan.fleck@statistik.gv.at

https://github.com/s-fleck