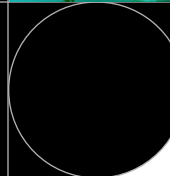
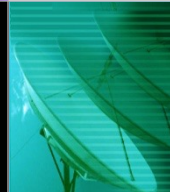


API in R : Plumber

강동훈



API in R : Plumber

Donghoon Gang



Contents



- Introduction
- What is REST?
- R-based REST API
- Server Implements
- Request and Response
- Reference

Introduction

- 데이터 분석에 대한 수요 증가
- R, Python등의 무료 오픈 소스 프로그래밍 언어를 데이터 분석에 활용



Introduction






- R 언어는 통계학자들이 주도하여 발전한 언어임
- 데이터 처리, 분석, 시각화에 특화
- Python에 비해 어플리케이션화 하여 활용하기가 어려움

빅데이터 인력양성 체계 마련을 위한 빅데이터 직무분석					
핵심능력	능력단위				
A 기획	A1 환경 분석	A2 분석 과제 정의	A3 분석 로드맵 수립	A4 성과 관리	
B 처리	B1 수집	B2 저장	B3 추출	B4 통합	B5 사전분석
C 분석	C1 요건 정의	C2 모델링 (알고리즘)	C3 검증 및 테스트	C4 적용	
D 시각화	D1 시각화 기획	D2 시각화 모델링	D3 시각화 디자인	D4 시각화 구축	D5 시각화 배포 및 유지보수
E 운영관리	E1 시스템 분석/운영	E2 용량/비용 계획	E3 보안 관리 (Security)		

출처 : 한국데이터베이스진흥원

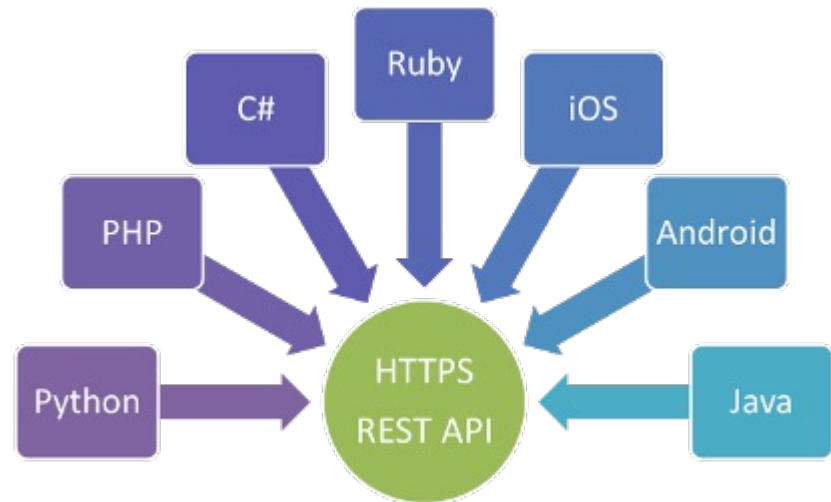
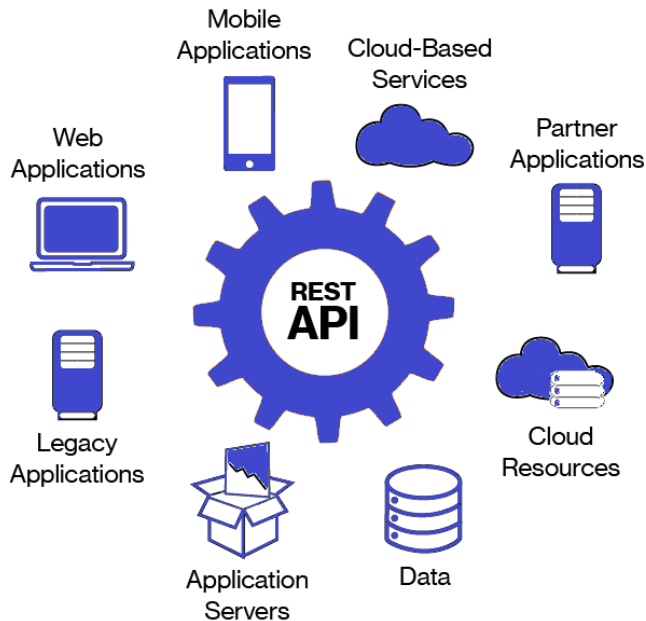
Introduction

- 어플리케이션으로 활용하기 위한 많은 시도들이 있었음
- R 기반의 web API

	프로젝트명	설명
	openCPU	클라우드 서버 호스팅 기반의 HTTP API 대용량 과학계산 컴퓨팅에 사용
	deployR	마이크로소프트에서 개발 azure cloud 의 Machine Learning package
	rApache	아파치 서버에서 구동하는 R 기반 웹 프로젝트 개발중단
	JUG	경량화된 R 기반 웹 개발 프레임워크 개발중단
	Plumber	R 코드 기반 web API 패키지 R studio team 에서 관리 2015년 7월 Rapier 에서 Plumber 로 이름 변경

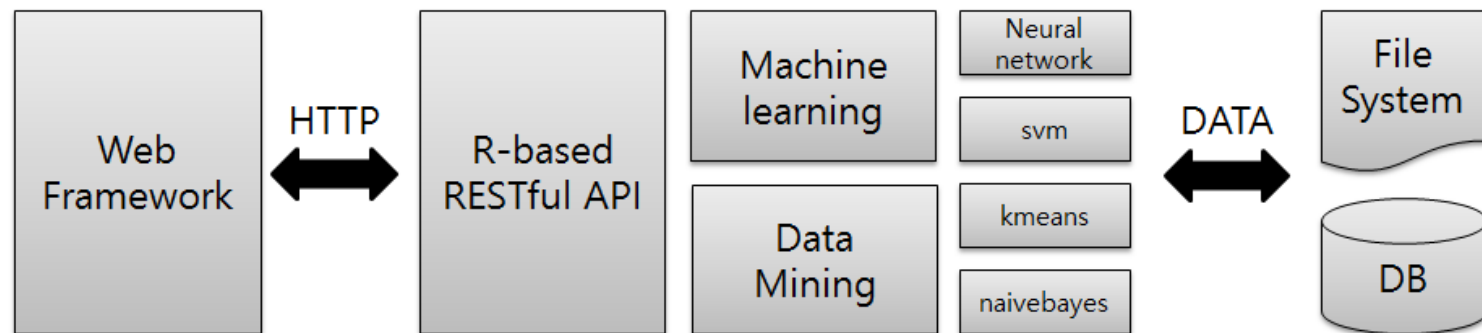
What is REST?

- 웹의 창시자 Roy Fielding의 논문에서 소개
- 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처
- REST API는 REST 아키텍처를 따르는 데이터 전송 방법
- 플랫폼, 프로그래밍 언어에 종속되지 않음



R-based REST API

- R 기반의 REST API 서비스 개념도
 - 알고리즘의 파라미터와 함께 URL로 서비스를 요청
 - 파일과 DB에서 데이터 조회
 - 데이터 처리
 - R 패키지에서 구현된 알고리즘을 활용
 - HTTP실행 결과 반환
 - AngularJS, Node.JS 등의 Web Framework 에서 결과 출력



R기반 REST API의 스키마 다이어그램

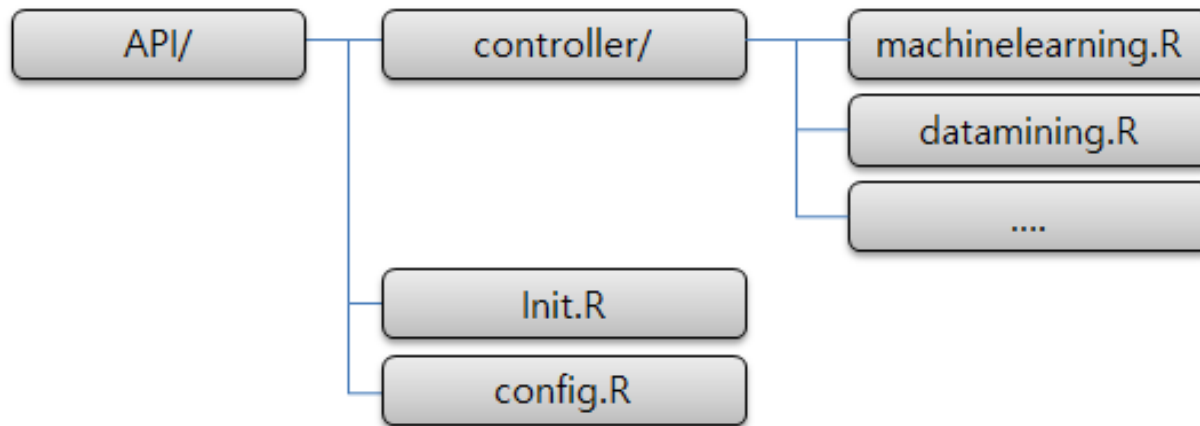
R-based REST API

- R 기반의 REST API 서비스 개념도
 - REST 요소 정의
 - 메서드, 리소스, 메시지 정의
 - URI 패턴 정의

메서드	리소스	메세지	URI
GET	machinelearning/nnet/	epoch, 활성화 함수	GET machinelearning/nnet/<epoch>/<actFunc>
GET	machinelearning/svm/	감마, 커널 함수	GET machinelearning/svm/<gamma>/<kernel>
GET	datamining/kmeans/	클러스터 개수	GET datamining/kmeans/<ncluster>
GET	datamining/naivebayes/	(없음)	GET datamining/naivebayes

Server Implements

- 효율적인 코드 관리를 위한 폴더 구조
- 리소스는 controller/ 폴더에서 관리
- init.R 파일로 서버 구동
- config.R 파일에서 전역변수 및 설정 값을 모아 관리



API 코드 폴더구조

Server Implements

- 최 상위 루트 객체 생성
- Controller/ 폴더 내 리소스 파일 읽음
- 최 상위 루트 객체 하위에 리소스 추가 (mount)
- 서버 실행

```
1 library(plumber) # server
2 library(RMySQL) # DB
3 CONTROLLER_PATH = "controller/"
4 server <- plumber$new()
5 controllers <- gsub( ".R" , "" , list.files( path = CONTROLLER_PATH ) )
6 for (controller in controllers) {
7   m <- plumber$new( paste( CONTROLLER_PATH , controller , ".R" , sep = "" ) )
8   server$mount( paste("/", controller, sep="") , m)
9 }
10 server$run(port=8000, host = "0.0.0.0")
```

Init.R 코드 예제

Server Implements

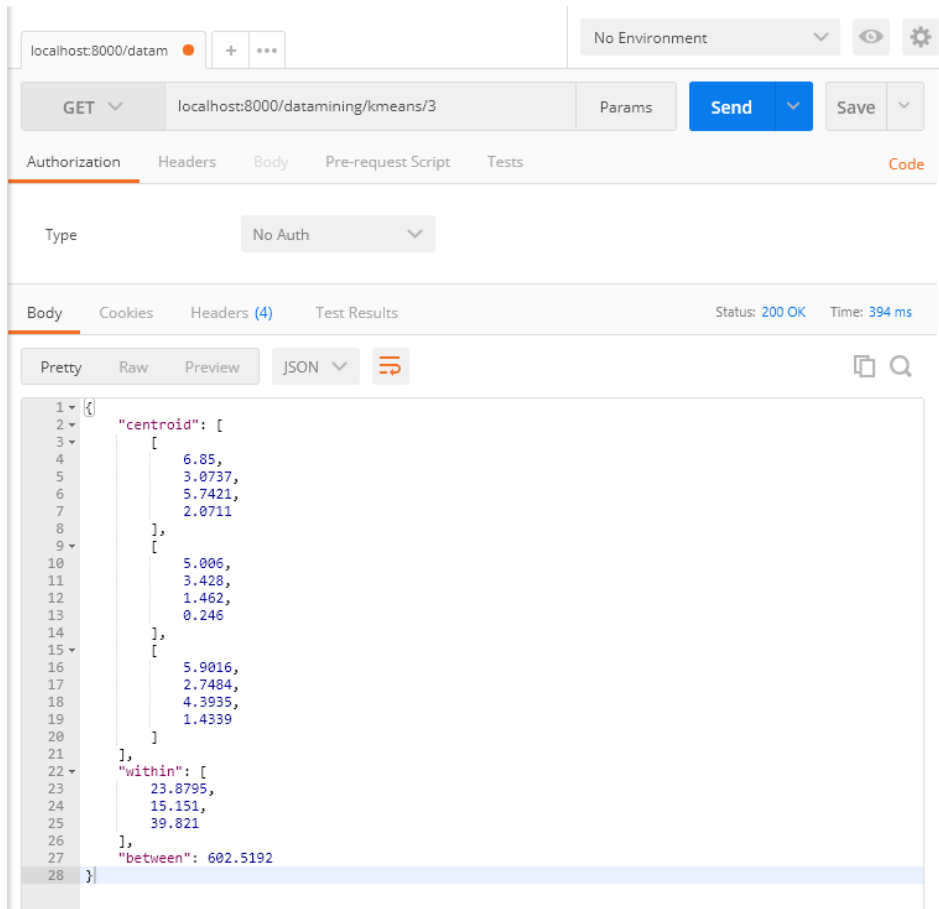
- 리소스 코드 구현
- Annotation(의미가 있는 주석) 을 이용하여 controller 매핑
 - 리턴타입
 - 하위주소의 End-Point
 - 메시지 패턴 설정

```
1  /** @serializer unboxedJSON # return JSON string
2  /** @get /kmeans/<nCluster> # resource subpath endpoint
3  kmeansCluster <- function(nCluster){
4  model <- kmeans(iris[,1:4], centers = nCluster)
5  return (list(centroid=model$centers, within=model$withinss, between=model$betweenss))
6  }
```

Datamining.R 리소스 코드 예제

Response and Request

- 서버 터미널에서 init.R 실행
- "localhost:8000/datamining/kmeans/3" 주소로 API 호출



The screenshot shows a REST client interface with the URL `localhost:8000/datamining/kmeans/3` and the method `GET`. The response status is `200 OK` and the time taken is `394 ms`. The response body is displayed in JSON format, showing the results of a Kmeans clustering operation.

```
1 {
2   "centroid": [
3     [
4       6.85,
5       3.0737,
6       5.7421,
7       2.0711
8     ],
9     [
10      5.006,
11      3.428,
12      1.462,
13      0.246
14    ],
15    [
16      5.9016,
17      2.7484,
18      4.3935,
19      1.4339
20    ]
21  ],
22   "within": [
23     23.8795,
24     15.151,
25     39.821
26  ],
27   "between": 602.5192
28 }
```

Kmeans API 요청에 대한 응답결과

Summary

- 요약
 - R의 Plumber 패키지를 이용하여 REST API를 구현
 - API 요구사항 확장에 따른 폴더와 파일 구조 제안
 - R 언어 기반의 서버 측 코드 작성 예제
- 단점 및 한계
 - API 보안
 - API 사용 권한
 - 패키지 의존성 관리
 - 대량의 유저를 위한 서비스에는 부적합
- R package
 - Plumber : <https://www.rplumber.io/>



Thank you