

Module: API-KMA forecast

LearningSpoonsR

2018-05-13

동네예보조회 (Internet Explorer를 사용하세요.) (<https://www.data.go.kr/subMain.jsp?param=T1BFTkFQSUAxNTAwMDA5OQ==#/L3B1YnlvcG90L215cC9Jcm9zTXlQYWdlL29wZW5EZXZhdWlkZVBhZ2UkQF4wMTJtMSRAXnB1Ym90RhdGFQZQZ>)

1. Query 준비

1.서비스 개요

1.1. 오퍼레이션 목록

서비스명 (국문)	오퍼레이션명(영문)	오퍼레이션명(국문)	메시지명(영문)
동네예보정보조회서비스 (최근 24 시간 자료)	getForecastGrib	초단기실황조회	getForecastGribRequest
	getForecastTimeData	초단기예보조회	getForecastTimeDataRequest
	getForecastSpaceData	동네예보조회	getForecastSpaceDataRequest
	getForecastVersionCheck	예보버전조회	getForecastVersionCheckRequest

[그림 1] 활용가이드.hwp 스크린샷

위의 [그림1] 처럼 4가지의 오퍼레이션을 제공합니다.

(3) 동네예보조회 오퍼레이션 명세

오퍼레이션 정보	오퍼레이션 번호	3	오퍼레이션명(국문)	동네예보조회
	오퍼레이션 유형	조회(상세)	오퍼레이션명(영문)	getForecastSpaceData
	오퍼레이션 설명	동네예보 정보를 조회하기 위해 발표일자, 발표시각, 예보지점 X 좌표, 예보지점 Y 좌표의 조회 조건으로 발표일자, 발표시각, 자료구분문자, 예보값, 예보일자, 예보시각, 예보지점 X 좌표, 예보지점 Y 좌표의 정보를 조회하는 기능		
	Call Back URL	N/A		
	최대 메시지 사이즈	[48452 bytes]		
	평균 응답 시간	[547ms]		
	호출 메시지정보	메시지명(영문) getForecastSpaceDataRequest 메시지타입 변수형 메시지설명 검색구분, 검색어 해당하는 동네예보 정보를 조회한다. 선행 오퍼레이션 N/A		
응답메시지 정보	메시지명(영문)	getForecastSpaceDataResponse		
	메시지 타입	리스트형		
	메시지 설명	호출 메시지의 검색 조건에 따라 단기예보 정보를 제공한다.		
HTTP Method		[O] REST (GET, POST, PUT, DELETE)		

[그림 2] 활용가이드.hwp 스크린샷

위의 [그림2]은 오퍼레이션의 개요를 설명합니다.

요청 메시지 명세

메시지명	http://newsky2.kma.go.kr/service/SecndSrtpdFrstInfoService2/ForecastSpaceData				
항목명	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
(필수) ServiceKey	서비스 키	255	1	TEST_SERVICE_KEY	서비스 인증
(필수) base_date	발표일자	8	1	20151201	'15년 12월 1일 발표
(필수) base_time	발표시각	4	1	0500	05시 발표 * 하단 참고자료 참조
(필수) nx	예보지점 X 좌표	2	0	1	예보지점의 X 좌표값
(필수) ny	예보지점 Y 좌표	2	0	1	예보지점의 Y 좌표값
numOfRows	한 페이지 결과 수	2	0	10	한 페이지 결과 수
pageNo	페이지 번호	5	0	1	페이지 번호
_type	타입			xml, json	xml(기본값), json

[그림 3] 활용가이드.hwp 스크린샷

위의 요청메시지를 작성하여 API database에 접속합니다. 아래의 코드블럭으로 이를 구성해보겠습니다.

```
svc_key <- paste0("uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig41J7iHwIE5w",
  "GS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D")
today <- gsub("-", "", Sys.Date())
url <- paste0("http://newsky2.kma.go.kr/service/SecndSrtpdFrstInfoService2",
  "/ForecastSpaceData")
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type")
values <- c(svc_key, today, "0800", "60", "127", "json")
request <- paste(fields, values, sep = "=") %>% paste(collapse="&")
query <- paste0(url, "?", request)
cat(query)
```

```
## http://newsky2.kma.go.kr/service/SecndSrtpdFrstInfoService2/ForecastSpaceData?ServiceKey=uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig41J7iHwIE5wGS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D&base_date=20180513&base_time=0800&nx=60&ny=127&_type=json
```

request의 fields와 values는 메뉴얼과 웹사이트를 참조하여 구성해야 합니다. 특히 nx와 ny는 우리나라의 특정 지점의 좌표를 지정해주는 명령어입니다. nx=60, ny=127은 서울특별시, 혹은 서울특별시의 종로구와 중구의 좌표입니다.

	C	D	E	F	G
1	1단계	2단계	3단계	격자 X	격자 Y
2	서울특별시			60	127
3	서울특별시	종로구		60	127
4	서울특별시	종로구	청운효자동	60	127
5	서울특별시	종로구	사직동	60	127
6	서울특별시	종로구	삼청동	60	127
7	서울특별시	종로구	부암동	60	127
8	서울특별시	종로구	평창동	60	127
9	서울특별시	종로구	무악동	60	127
10	서울특별시	종로구	교남동	60	127
11	서울특별시	종로구	가회동	60	127
12	서울특별시	종로구	종로1.2.3.4가동	60	127
13	서울특별시	종로구	종로5.6가동	60	127
14	서울특별시	종로구	이화동	60	127
15	서울특별시	종로구	혜화동	60	127
16	서울특별시	종로구	창신제1동	61	127

[그림 4] 동네예보조회서비스_격자_위경도.xls 스크린샷

url과 request를 조합하여 query가 작성되었습니다. 위의 query를 웹브라우저에 붙여넣어서 결과를 확인할 수 있습니다.

➤ 요청/응답 메시지 명세(예시)

REST(URI)
http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData?base_date=20151021&base_time=0230&nx=1&ny=1
http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData?base_date=20151021&base_time=0230&nx=1&ny=1&_type=json

[그림 5] 활용가이드.hwp 스크린샷

메뉴얼에서도 위의 [그림5]처럼 예시를 제공하고 있습니다.

➤ 응답 메시지 명세

메시지명	getForecastSpaceDataResponse				
항목명(영문)	항목명(국문)	항목 크기	항목 구분	샘플데이터	항목설명
resultCode	결과코드	4	1	0000	결과코드
resultMsg	결과메시지	50	1	OK	결과메시지
numOfRows	한페이지 결과 수	2	0	10	한 페이지 결과 수
pageNo	페이지 번호	5	0	1	페이지 번호
totalCount	전체 결과 수	7	0	12334	전체 결과 수
baseDate	발표일자	8	1	20151201	'15년 12월 1일 발표
baseTime	발표시각	6	1	0500	05시 발표
fcstDate	예보일자	8	1	20151201	'15년 12월 1일 예보
fcstTime	예보시각	4	1	0900	9시 예보
category	자료구분문자	3	1	POP	자료구분코드 (하단 코드값 정보 참조)
fcstValue	예보 값	2	0	-1	*T3H, TMN, TMX, UUU, VVV, WAV, WSD 실수로 제공 (하단 코드값 정보 참조)
nx	예보지점 X 좌표	2	0	5	예보지점 X 좌표
ny	예보지점 Y 좌표	2	0	2	예보지점 Y 좌표

[그림 6] 활용가이드.hwp 스크린샷

위의 [그림6]는 해당 query 를 통해서 얻을 수 있는 정보입니다.

2. 데이터 프레임으로 정리

JSON DATA

```

{"response":{"header":{"resultCode":"0000","resultMsg":"OK"},"body":{"items":{"item":{"baseDate":"20151021","baseTime":"0500","category":"T3H","fcstDate":"20151021","fcstTime":"0900","fcstValue":-50,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"UUU","fcstDate":"20151021","fcstTime":"0900","fcstValue":-5,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"VVV","fcstDate":"20151021","fcstTime":"0900","fcstValue":-1,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"POP","fcstDate":"20151021","fcstTime":"0900","fcstValue":-1,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"REH","fcstDate":"20151021","fcstTime":"0900","fcstValue":-1,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"PTY","fcstDate":"20151021","fcstTime":"0900","fcstValue":0,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"R06","fcstDate":"20151021","fcstTime":"0900","fcstValue":0,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"S06","fcstDate":"20151021","fcstTime":"0900","fcstValue":0,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"TMN","fcstDate":"20151021","fcstTime":"0900","fcstValue":0,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"TMX","fcstDate":"20151021","fcstTime":"0900","fcstValue":0,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"SKY","fcstDate":"20151021","fcstTime":"0900","fcstValue":1,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"WAV","fcstDate":"20151021","fcstTime":"0900","fcstValue":1,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"WSD","fcstDate":"20151021","fcstTime":"0900","fcstValue":5,"nx":1,"ny":1},
{"baseDate":"20151021","baseTime":"0500","category":"VEC","fcstDate":"20151021","fcstTime":"0900","fcstValue":74,"nx":1,"ny":1},
{"numOfRows":308,"pageNo":1,"totalCount":308}}}}

```

[그림 7] 활용가이드.hwp 스크린샷

[그림7]는 json형식으로 조회를 했을때의 예상 결과입니다. 위의 query 를 실행하여 결과를 확인해보겠습니다.

```

raw <- readLines(query, warn = "F", encoding = "UTF-8")
print(raw)

```

```

## [1] "{\"response\":{\"header\":{\"resultCode\":\"0000\",\"resultMsg\":\"OK\"},\"body\":{\"items\":{\"item\":{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"POP\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":10,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"PTY\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":0,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"R06\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":65,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"REH\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":65,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"S06\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":0,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"SKY\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":2,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"T3H\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":20,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"UUU\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":2.9,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"VEC\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":295,\"nx\":60,\"ny\":127},{\"baseDate\":\"20180513\",\"baseTime\":\"0800\",\"category\":\"VVV\",\"fcstDate\":\"20180513\",\"fcstTime\":\"1200\",\"fcstValue\":-1.4,\"nx\":60,\"ny\":127}}},\"numOfRows\":10,\"pageNo\":1,\"totalCount\":216}}}"

```

데이터는 잘 불러져 왔습니다. json형식의 data를 data.frame 으로 바꾸는 일만 남았습니다. (지금은 데이터의 이름이 raw 이지만, data.frame 이 되면 dataset 이라는 이름을 붙여줄 예정입니다.)

```

activate("jsonlite")
raw <- fromJSON(raw)
str(raw)

```

```
## List of 1
## $ response:List of 2
## ..$ header:List of 2
## .. ..$ resultCode: chr "0000"
## .. ..$ resultMsg : chr "OK"
## ..$ body :List of 4
## .. ..$ items :List of 1
## .. .. ..$ item:'data.frame': 10 obs. of 8 variables:
## .. .. ..$ baseDate : int [1:10] 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513
## .. .. ..$ baseTime : chr [1:10] "0800" "0800" "0800" "0800" ...
## .. .. ..$ category : chr [1:10] "POP" "PTY" "R06" "REH" ...
## .. .. ..$ fcstDate : int [1:10] 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513 20180513
## .. .. ..$ fcstTime : int [1:10] 1200 1200 1200 1200 1200 1200 1200 1200 1200 1200
## .. .. ..$ fcstValue: num [1:10] 10 0 0 65 0 2 20 2.9 295 -1.4
## .. .. ..$ nx : int [1:10] 60 60 60 60 60 60 60 60 60 60
## .. .. ..$ ny : int [1:10] 127 127 127 127 127 127 127 127 127 127
## ..$ numOfRows : int 10
## ..$ pageNo : int 1
## ..$ totalCount: int 216
```

이제 거의 다 되었습니다. raw 라는 객체가 조금씩 모습을 잡아가고 있습니다.

```
dataset <- raw$response$body$items$item
dataset
```

	baseDate <int>	baseTime <chr>	category <chr>	fcstDate <int>	fcstTime <int>	fcstValue <dbl>	nx <int>	ny <int>
1	20180513	0800	POP	20180513	1200	10.0	60	127
2	20180513	0800	PTY	20180513	1200	0.0	60	127
3	20180513	0800	R06	20180513	1200	0.0	60	127
4	20180513	0800	REH	20180513	1200	65.0	60	127
5	20180513	0800	S06	20180513	1200	0.0	60	127
6	20180513	0800	SKY	20180513	1200	2.0	60	127
7	20180513	0800	T3H	20180513	1200	20.0	60	127
8	20180513	0800	UUU	20180513	1200	2.9	60	127
9	20180513	0800	VEC	20180513	1200	295.0	60	127
10	20180513	0800	VVV	20180513	1200	-1.4	60	127

1-10 of 10 rows

드디어 dataset 을 만들어 냈습니다.

3. Get the full record!

그런데 과연 우리가 확보한 데이터가 전부인가요?

```
raw$response$body[-1]
```

```
## $numOfRows
## [1] 10
##
## $pageNo
## [1] 1
##
## $totalCount
## [1] 216
```

저희는 10개의 관찰값을 확보했는데, 이것은 pageNo 1에 해당합니다. 그런데 해당 request 에 대해서 실제 data의 갯수는 216입니다. 즉, 저희는 1번-10번 관찰값만 확보를 하였고, 11번-216번 관찰값은 확보하지 못했습니다.

query 를 작성했던 코드블럭과 [그림3]을 다시한번 보겠습니다.

➤ 요청 메시지 명세

메시지명	http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData				
항목명	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
(필수)ServiceKey	서비스 키	255	1	TEST_SERVICE_KEY	서비스 인증
(필수)base_date	발표일자	8	1	20151201	'15년 12월 1일 발표
(필수)base_time	발표시각	4	1	0500	05시 발표 * 하단 참고자료 참조
(필수)nx	예보지점 X 좌표	2	0	1	예보지점의 X 좌표값
(필수)ny	예보지점 Y 좌표	2	0	1	예보지점의 Y 좌표값
numOfRows	한 페이지 결과 수	2	0	10	한 페이지 결과 수
pageNo	페이지 번호	5	0	1	페이지 번호
_type	타입			xml, json	xml(기본값), json

[그림 8] 활용가이드.hwp 스크린샷

```
svc_key <- paste0("uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig4lJ7iHwIE5w",
                  "GS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D")
today <- gsub("-", "", Sys.Date())
url <- paste0("http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2",
              "/ForecastSpaceData")
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type")
values <- c(svc_key, today, "0800", "60", "127", "json")
request <- paste(fields, values, sep = "=") %>% paste(collapse="&")
query <- paste0(url, "?", request)
cat(query)
```

fields 에 pageNo 를 1부터 22로 바꿔가면서 실행하고 그 결과를 합해야 한다는 것이 보이시나요? 이런 반복문을 만드는 작업을 해보겠습니다.

i 라는 변수가 1,2,3,...,22의 차례로 바뀌어야 합니다. i 가 바뀔 때마다 pageNo 에 해당하는 values 가 바뀌어야 하고, 그 결과로 query 가 바뀌어야 합니다.

Q. 여기에서 22라는 값은 왜 22이죠?

- A. 22*10이면 216보다 크거나 같기 때문입니다.
- raw\$response\$body\$totalCount = 216
- raw\$response\$body\$numOfRows = 10
- ceiling(raw\$response\$body\$totalCount/raw\$response\$body\$numOfRows)

```
maxPage <- ceiling(raw$response$body$totalCount/raw$response$body$numOfRows)
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type", "pageNo")
for (i in 1:maxPage) {
  print(i)
  values_i <- c(svc_key, today, "0800", "60", "127", "json", i)
  request_i <- paste(fields, values_i, sep = "=") %>% paste(collapse="&")
  query_i <- paste0(url, "?", request_i)
  if (i %in% c(5,10,15)) cat(query_i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData?ServiceKey=uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig4lJ7iHwIE5wGS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D&base_date=20180513&base_time=0800&nx=60&ny=127&_type=json&pageNo=5[1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData?ServiceKey=uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig4lJ7iHwIE5wGS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D&base_date=20180513&base_time=0800&nx=60&ny=127&_type=json&pageNo=10[1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2/ForecastSpaceData?ServiceKey=uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFEig4lJ7iHwIE5wGS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D&base_date=20180513&base_time=0800&nx=60&ny=127&_type=json&pageNo=15[1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
```

query_i 가 구성되었습니다. 이제 query_i 를 $i=1,2,3,\dots,maxPage$ 에 대해서 구성하고 이를 반복해서 dataset_i 를 만들고 이것을 합치는 일만 남았습니다. 위의 블록에서 부터 어떤 내용이 추가되어야 할까요?

```
maxPage <- ceiling(raw$response$body$totalCount/raw$response$body$numOfRows)
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type", "pageNo")

dataset <- data.frame()
for (i in 1:maxPage) {
  values_i <- c(svc_key, today, "0800", "60", "127", "json", i)
  request_i <- paste(fields, values_i, sep = "=") %>% paste(collapse="&")
  query_i <- paste0(url, "?", request_i)

  raw_i <- readLines(query_i, warn = "F", encoding = "UTF-8")
  raw_i <- fromJSON(raw_i)
  dataset_i <- raw_i$response$body$items$item
  dataset <- rbind(dataset, dataset_i)
}
```

이제 dataset 이 완성되었습니다. rmarkdown html output에서의 꽤 괜찮은 presentation을 위해서 kable 과 kableExtra 패키지를 사용해서 출력해보겠습니다.

```
activate(c("kableExtra", "knitr"))
kable(head(dataset)) %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                position = "center")
```

```
## Warning in kable_styling(., bootstrap_options = "striped", full_width =
## F, : Please specify format in kable. kableExtra can customize either HTML
## or LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

baseDate	baseTime	category	fcstDate	fcstTime	fcstValue	nx	ny
20180513	0800	POP	20180513	1200	10	60	127
20180513	0800	PTY	20180513	1200	0	60	127
20180513	0800	R06	20180513	1200	0	60	127
20180513	0800	REH	20180513	1200	65	60	127
20180513	0800	S06	20180513	1200	0	60	127
20180513	0800	SKY	20180513	1200	2	60	127

4. 실제 사용시의 코드

위의 코드들은 순차적인 설명을 위해서 단계별로 작성한 것입니다. 이를 정리해서 한번에 실행하는 코드의 예제를 아래에 보여드리겠습니다.

```
# 0. setup environment
source("LSR.R")
activate(c("dplyr", "jsonlite"))

# 1. setup API
svc_key <- paste0("uEID6no5W0eFLEu%2FYZpdjKHQVrE2HtFeig4lJ7iHwIE5w",
                  "GS1L3RvmusPMDkumoj8f%2BSffvPYW0%2B5xXu%2FrQ%2Bvzg%3D%3D")
today <- gsub("-", "", Sys.Date())
url <- paste0("http://newsky2.kma.go.kr/service/SecndSrtpdFrcstInfoService2",
              "/ForecastSpaceData")

# 2. get maxPage
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type")
values <- c(svc_key, today, "0800", "60", "127", "json")
request <- paste(fields, values, sep = "=") %>% paste(collapse="&")
query <- paste0(url, "?", request)
raw <- readLines(query, warn = "F", encoding = "UTF-8") %>% fromJSON()
maxPage <- ceiling(raw$response$body$totalCount/raw$response$body$numOfRows)

# 3. collect all pages
fields <- c("ServiceKey", "base_date", "base_time", "nx", "ny", "_type", "pageNo")
dataset <- data.frame()
for (i in 1:maxPage) {
  values_i <- c(svc_key, today, "1700", "60", "127", "json", i)
  request_i <- paste(fields, values_i, sep = "=") %>% paste(collapse="&")
  query_i <- paste0(url, "?", request_i)
  raw_i <- readLines(query_i, warn = "F", encoding = "UTF-8") %>% fromJSON()
  dataset_i <- raw_i$response$body$items$item
  dataset <- rbind(dataset, dataset_i)
}

# 4. deliver output
print(dataset)
```

5. Summary

