

프로그램 기능 목록

구분	기능번호	상세 spec	비고
지역 별 날씨 정보	Wea1	지역 별 날씨 정보를 제공. - 평균온도, 강수확률 등	
	Wea2	미세먼지 농도 정보 제공	
교통 정보 제공	Trans1	실시간 대중교통 정보 제공 -버스 : 번호, 도착예정시간	시외버스, 고속버스 시간표 및 실시간 고속도로 정보 제공 확대.
	Trans2	-지하철 : 방향, 도착 예정시간.	
개인 별 맞춤 정보	Pri1	1 : 1 맞춤형 서비스. 사용자가 즐겨 찾는 정보를 맞춤으로 제공.	
	Pri2	전체 사용자 기준. 많이 찾는 정보 제공.	

카카오톡 API에 필요한 Object

Keyboard

응답 메시지에 따라 사용자의 키보드 영역에 표현될 메시지 입력방식에 대한 정보
별도로 설정하지 않는 경우 text 타입이 기본으로 설정

직접 입력

```
{ "type": "text" }
```

버튼 입력

```
{ "type": "buttons", "buttons": [ "선택 1", "선택 2", "선택 3" ] }
```

Message

카카오톡 이용자가 명령어를 선택 혹은 입력했을 때 이용자에게 전송하는 응답 메시지
String, photo, MessageButton의 조합으로 이루어짐

```
{ "text": "안녕하세요.",  
  "photo": { "url": "https://hello.photo.src", "width": 640, "height": 480 },  
  "message_button": { "label": "반갑습니다.", "url": "http://hello.world.com/example" } }
```

MessageButton

응답 메시지의 말풍선 하단에 보여지는 링크버튼 정보

```
{ "label": "쿠폰확인하기", "url": "http://coupon.coupon.com/~~" }
```

Database Table 정의 – MongoDB 사용

User table - 개인별 맞춤형 추천 알람 기능에 참조할 테이블

Column list

user_key (primary key) 〈String〉 : 사용자 식별 정보

search_key 〈Number〉 : 검색 구분 (날씨정보 / 교통정보1 (버스) / 교통정보2 (지하철))

Content 〈String〉 : 검색 내용

(날씨정보의 경우 지역명, 교통정보1 (버스)의 경우 정거장 이름, 교통정보2 (지하철)의 경우 역이름

예시

```
User {  
  {"1234", 1, "용산구"},  
  {"5678", 2, "강남역 사거리"},  
  {"4567", 3, "이태원역"} };
```

최초 사용자 화면 (자동응답 명령어 목록 호출) - 시스템 구성도

사용자 요청

Method : GET

URL : http(s)://:server_url/keyboard

Content-Type : application/json; charset=utf-8



사용자

Response

Return Keyboard:

{ "type" : "buttons", "buttons" : ["날씨 정보", "교통 정보", "개인별 알람 및 추천 설정"] }

MainController

Keyboard initMenu()

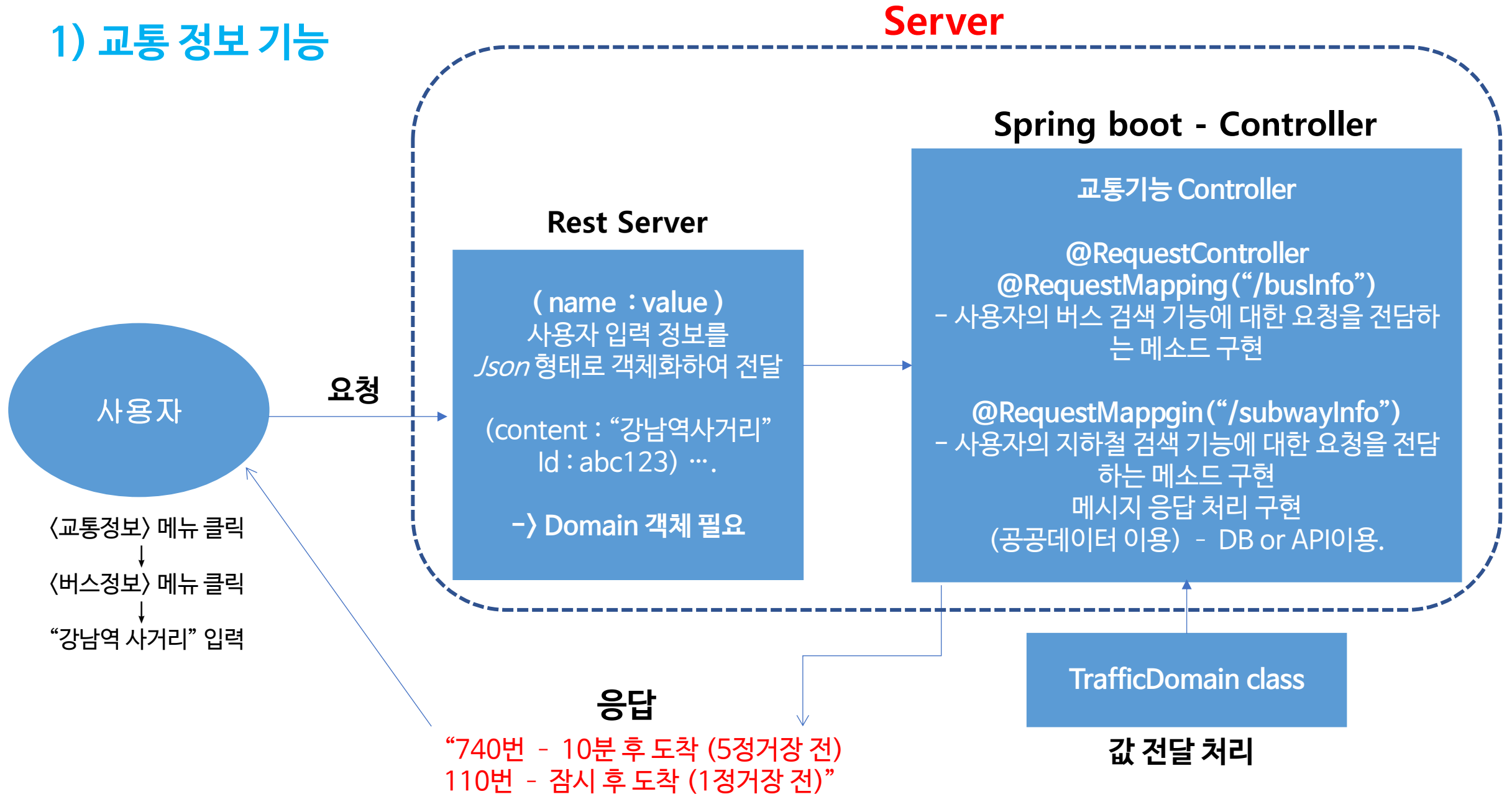
MainService

getMenuList()
InsertMenu()
deleteMenu()
modifyMenu()
... ..

MainMenu domain

String weatherInfo; // 날씨정보
String trafficInfo; // 교통정보
String customized; //개인 알람, 추천

1) 교통 정보 기능



1) 교통 정보 기능 - 시스템 구성도

사용자 request

Method : POST

URL : http(s)://:your_server_url/message

Content-Type : application/json; charset=utf-8

```
curl -XPOST 'https://:your_server_url/message' -d
```

```
{ "user_key": "1234",  
  "type": "text",  
  "content": "강남역사거리" }
```



사용자

“강남역 사거리” 입력

Response

```
{ "Message": { "text": "강남역 사거리 도착 정보: ....."} }
```

```
"Keyboard": { "type": "buttons", "buttons": [ "처음으로", "다른 검색", "즐거찾는 구간 추가" ] }
```

TrafficInfo domain

TrafficInfoController

```
busInfo(String user_key, String content)
subwayInfo(String user_key, String content)
```

TrafficInfoService

Dao.selectTrafficInfo()

TrafficInfoDao

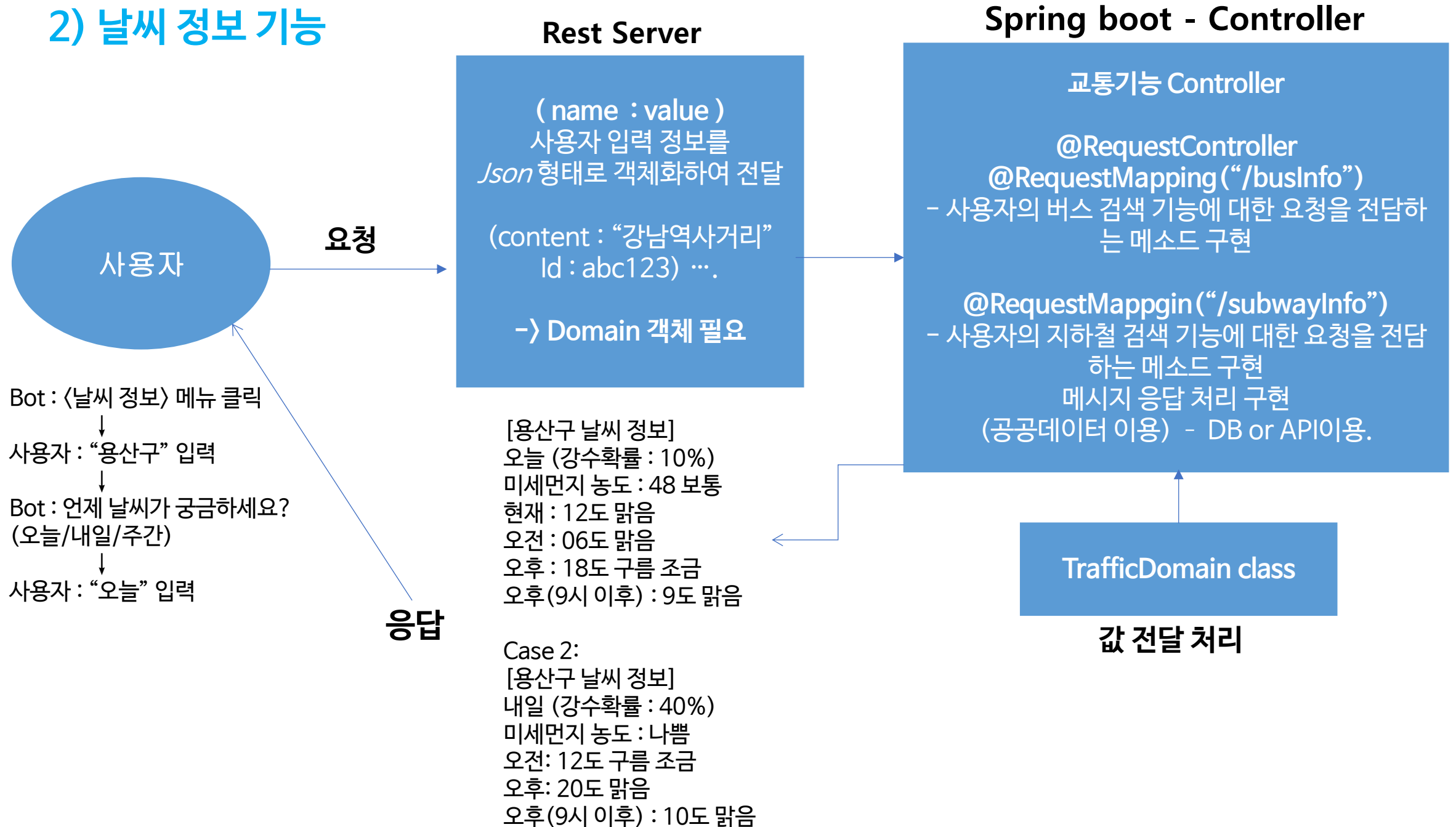
select

TrafficInfo - 공공데이터 API

API

(Member DB에 추가)

2) 날씨 정보 기능



2) 날씨 정보 기능 - 시스템 구성도

사용자 request

Method : POST

URL : http(s)://:your_server_url/message

Content-Type : application/json; charset=utf-8

```
curl -XPOST 'https://:your_server_url/message' -d
'{ "user_key": "1234",
  "type": "text",
  "content": "용산구" }'
```



사용자

“용산구” 입력

Response

```
{ "message": { "text": "용산구 날씨 정보" : ..... } }
"keyboard": { "type": "buttons", "buttons": [ "처음으로", "다른 검색", "즐거찾는 지역 추가" ] } }
```

(Member DB에 추가)

WeatherInfoController

WeatherInfo domain

searchInfo_today(String region)
searchInfo_week(String region)

WeatherInfoService

주요 메소드 정의

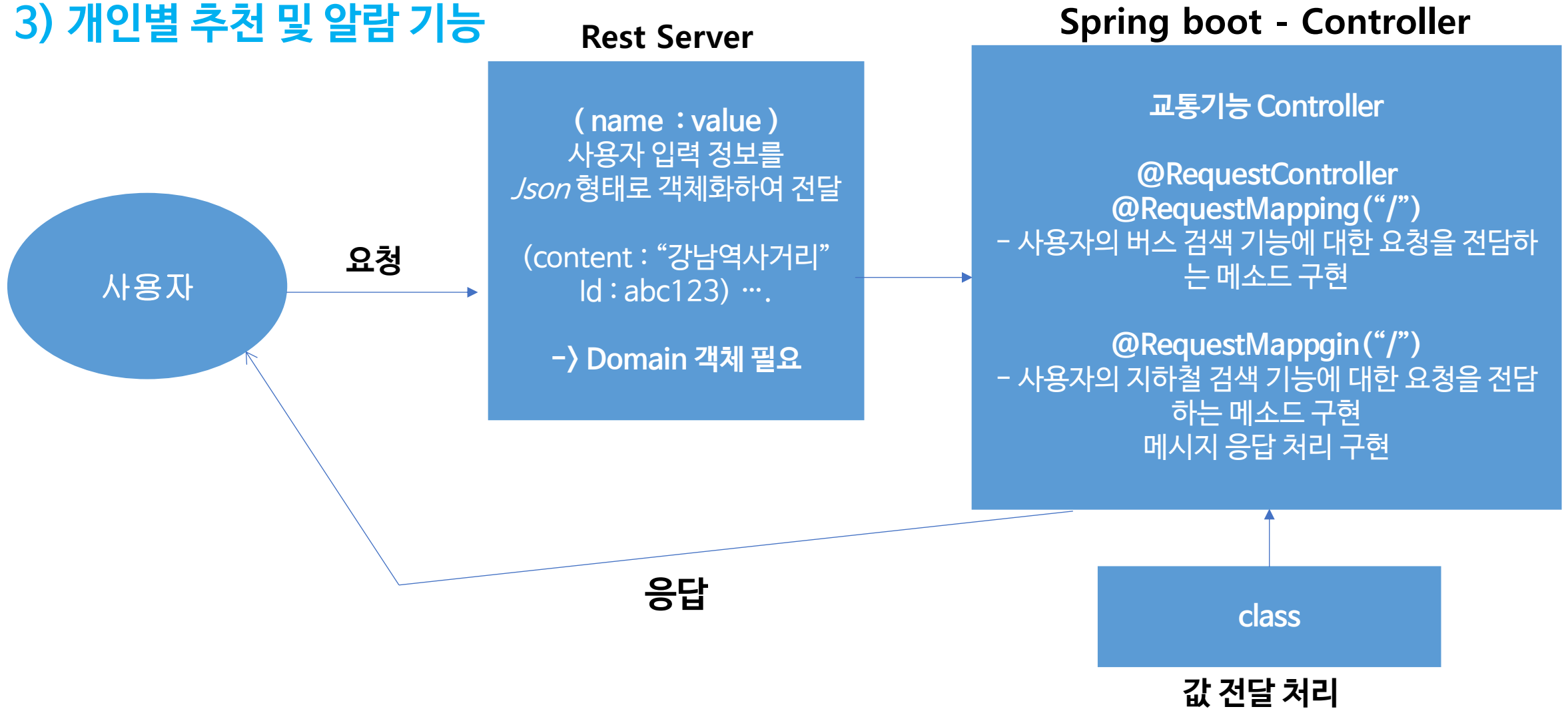
WeatherInfoDao

select

WeatherInfo - 공공데이터 API

API

3) 개인별 추천 및 알람 기능



※카카오 API를 통해 user_key를 response로 받아야 함

3) 개인별 추천 및 알람 기능 - 시스템 구성도

사용자 request

Method : POST

URL : http(s)://:your_server_url/message

Content-Type : application/json; charset=utf-8

```
curl -XPOST 'https://:your_server_url/message' -d
'{ "user_key": "1234",
  "type": "text",
  "content": "....." }'
```

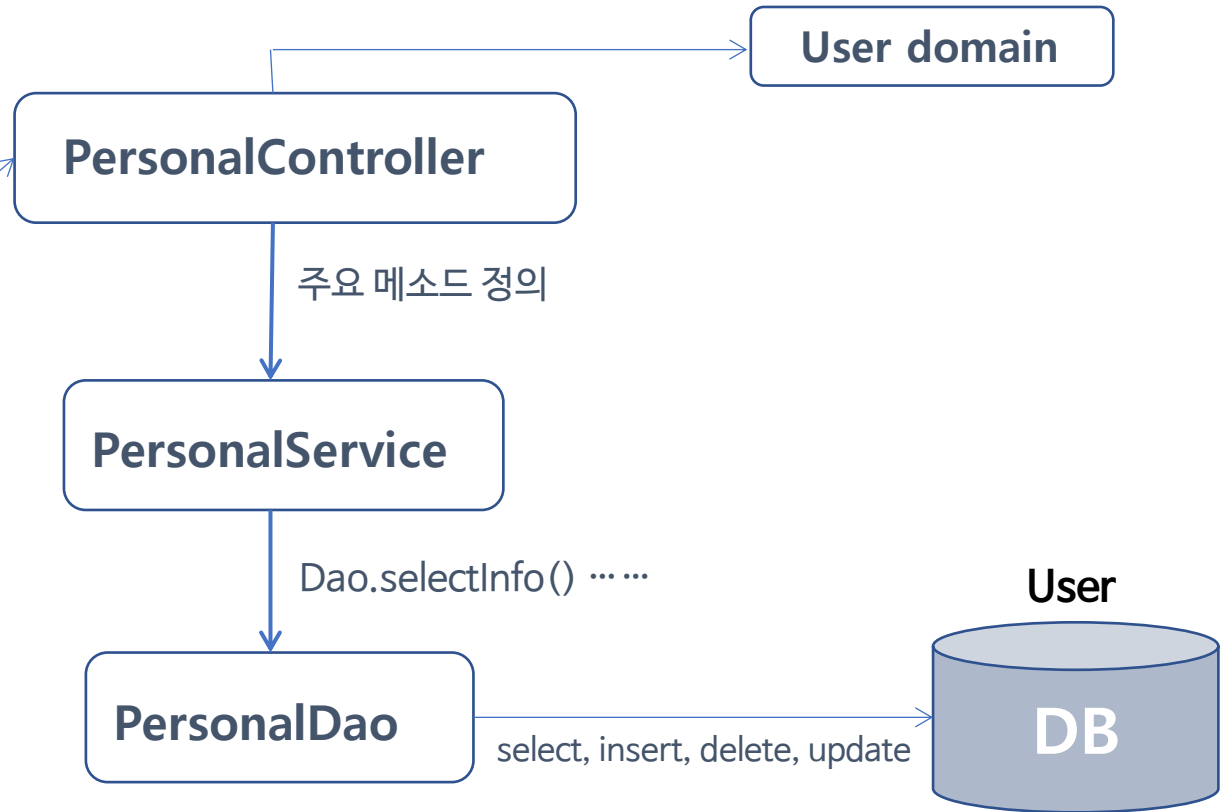


사용자

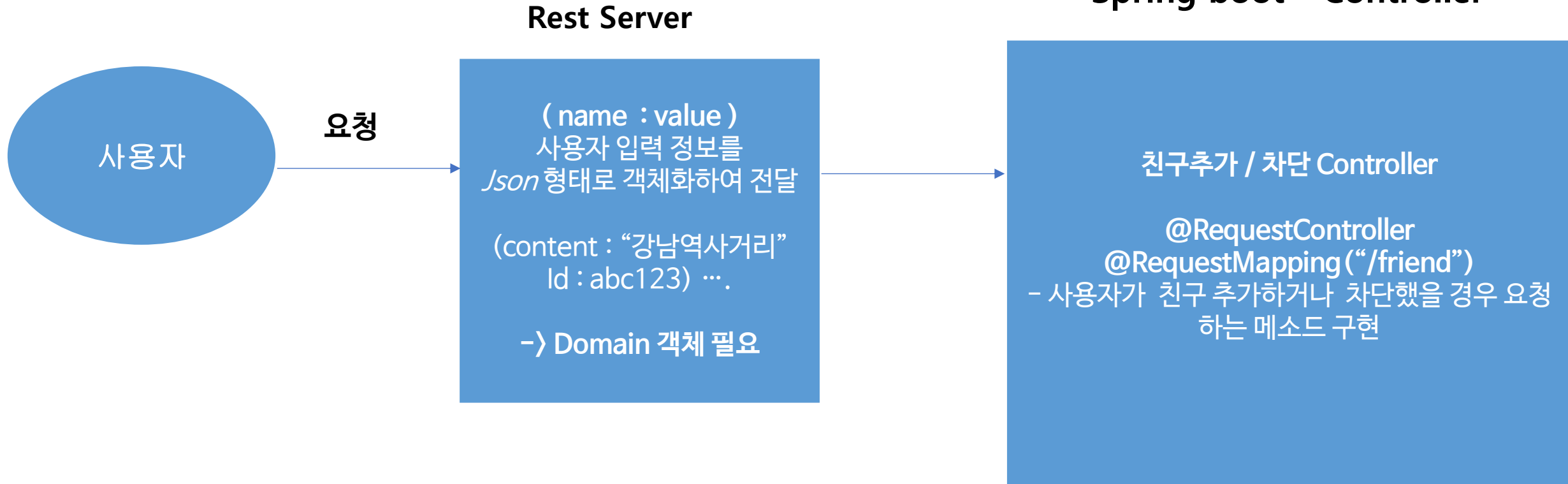
“강남역 사거리” 입력

Response

```
{ "message": { "text": "... .." } }
"keyboard": { "type": "buttons", "buttons": [ "...", "...", "..." ] }
```



4) 친구 추가/차단 알림 API



4) 친구 추가/차단 알림 API - 시스템 구성도

사용자 request

Method : POST /DELETE

URL : http(s)://:your_server_url/friend

Content-Type : application/json; charset=utf-8

친구 추가

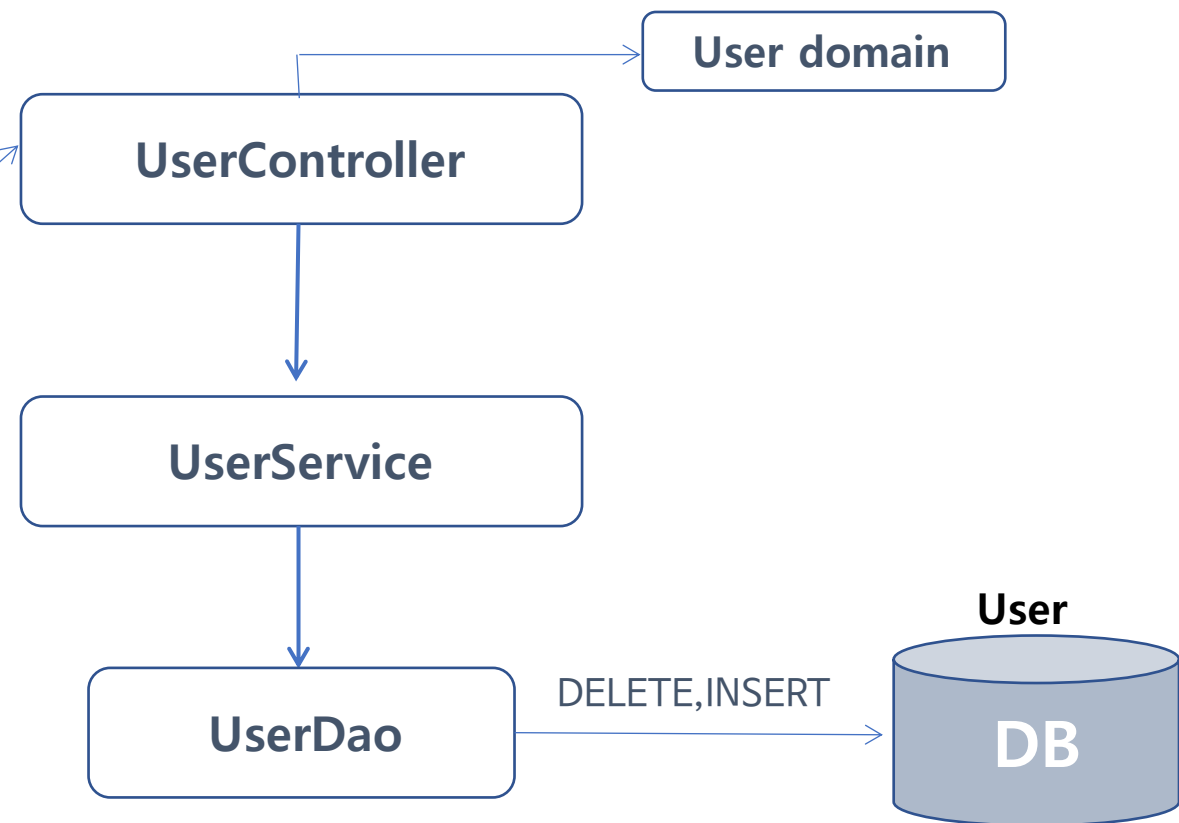
```
curl -XPOST 'https://:your_server_url/friend' -d  
'{"user_key" : "HASHED_USER_KEY" }'
```

친구 삭제

```
curl -XDELETE 'https://:your_server_url/friend/:user_key'
```



사용자
친구추가/ 차단



Response

http status code	code	message	comment
200	0	SUCCESS	정상 응답

5) 채팅방나가기 - 시스템 구성도

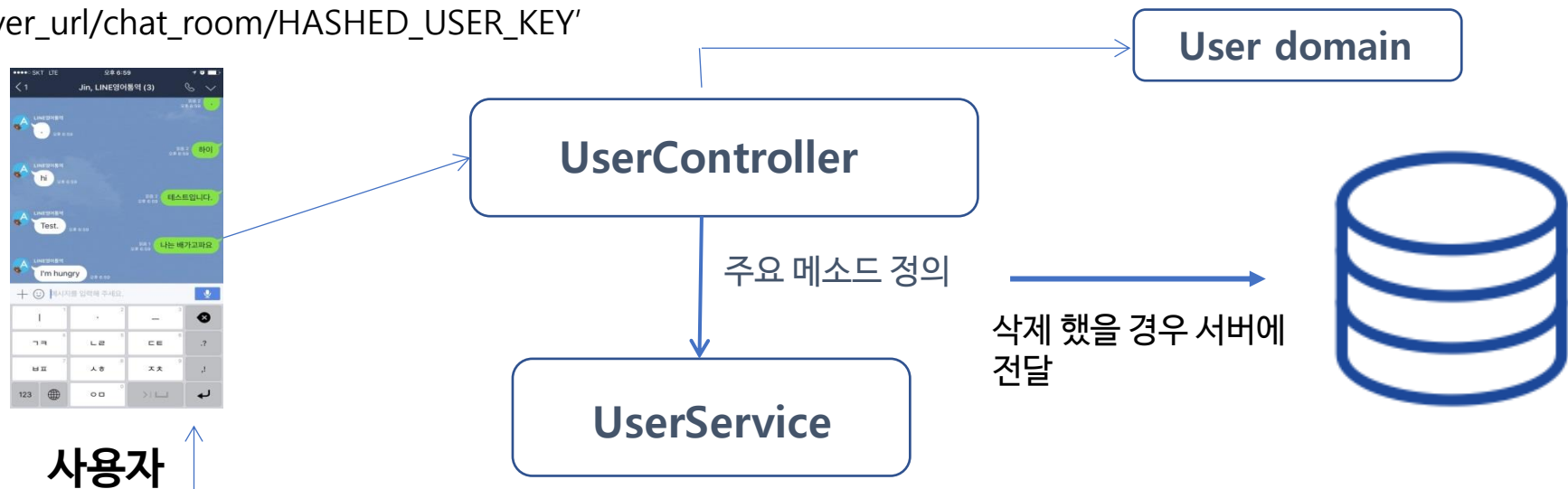
사용자 request

Method : DELETE

URL : http(s)://:your_server_url/chat_room/:user_key

Content-Type : application/json; charset=utf-8

curl -XPOST 'https://:your_server_url/chat_room/HASHED_USER_KEY'

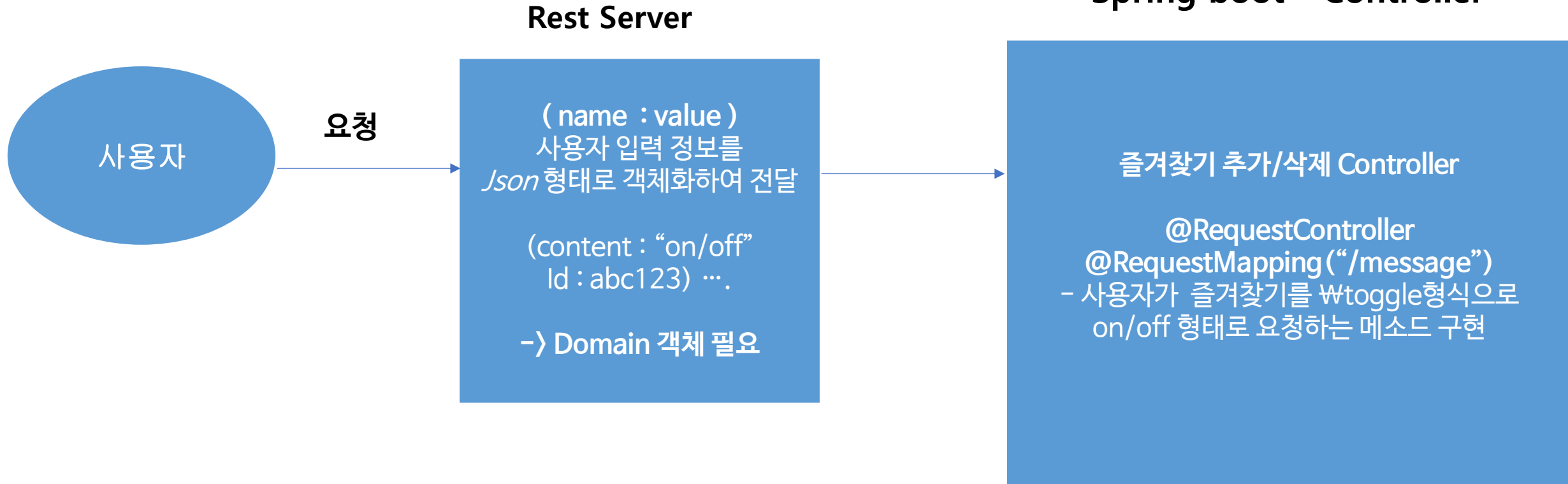


사용자

Response

http status code	code	message	comment
200	0	SUCCESS	정상 응답

6) 즐겨찾기 추가/삭제 알림 API



6) 즐겨찾기 추가 /삭제 API - 시스템 구성도

사용자 request

Method : POST /DELETE

URL : http(s)://:your_server_url/message

Content-Type : application/json; charset=utf-8

버튼 형식의 즐겨 찾기 버튼 추가

```
curl -XPOST 'https://:your_server_url/message'
```

```
{ "user_key" : "HASHED_USER_KEY"
```

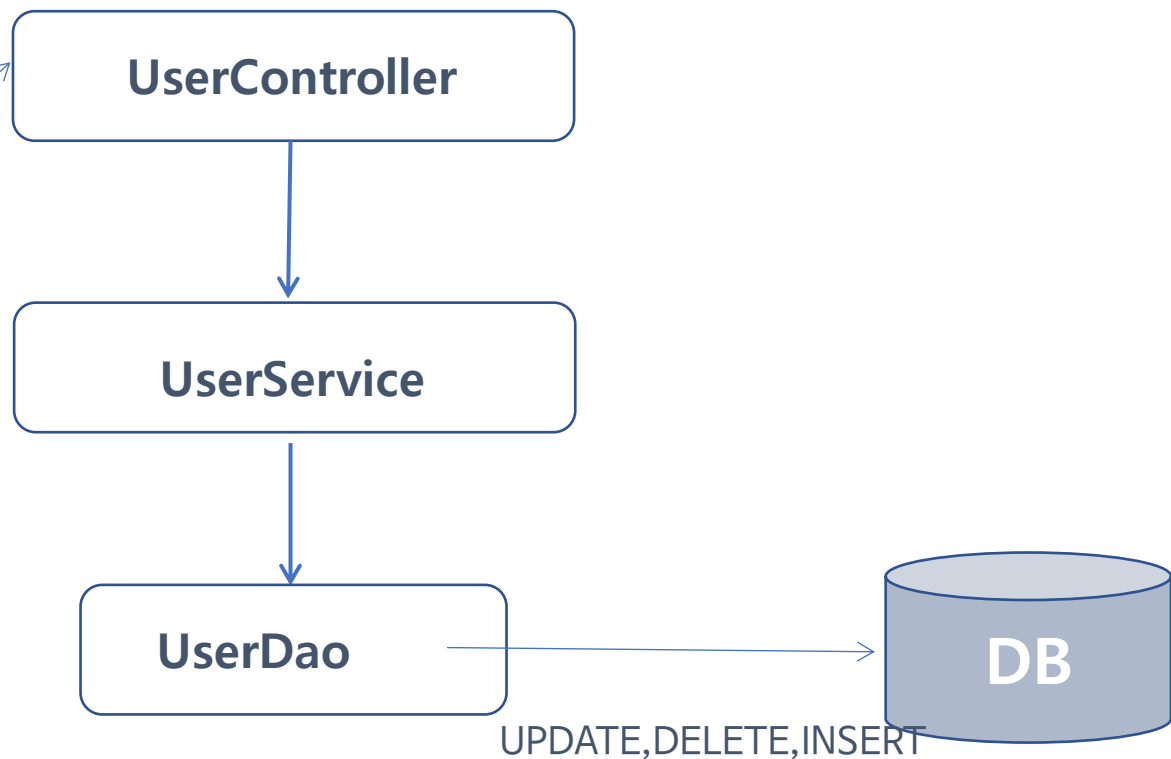
```
  "type": "onclick button",
```

```
  "content": "추가/삭제" }
```

즐거찾기 추가/삭제 버튼



사용자



기능 흐름도

