Basic of REST

HTTP URI + HTTP Method 이다. URI로 대상 자원을 명시하고 Method로 해당 자원에 대한 행위를 정의한다.

<http://bcho.tistory.com/321>

CoAp 도움될만한 사이트

<http://www.joinc.co.kr/modules/moniwiki/wiki.php/man/12/CoAP>

<http://www.epnc.co.kr/atl/view.asp?a_id=10881>

http://www.tta.or.kr/data/weekly\_view.jsp?news\_id=3110

Request body : Post/Put

Response body : GET

**Idempotent** : 멱등원. 반복되는 연산(method)에도 server의 상태는 바뀌지 않는다.(get, put, delete)

**새로운 리소스의 URI가 클라이언트에 의해 결정될 경우 PUT을 사용한다. 새로운 리소스의 URI가 서버에 의해 결정될 경우 POST를 사용한다**

Post는 부가된 리소스(자식)를 생성함으로 상태가 계속 바뀌지만(증가), put과 get, delete는 기존의 resource에 새로운 resource로 동작한다.

참조 : <http://greatkim91.tistory.com/14>

또는 반복되는 연산의 경우, 어떤 리소스를 DELETE하고 또 다시 DELETE 한다면 이미 지워진 내용을 또 지워도 결과는 같으므로 멱등이다. PUT으로 새 리소스를 생성하고 또 동일한 내용으로 PUT을 리소스에 요청해도 두 번째는 이미 존재하고 동일한 내용이므로 결과는 같다. 이미 존재하는 리소스에 PUT을 보낼 경우 동일한 내용으로 몇 번을 보내더라도 동일한 내용으로 상태가 저장되므로 리소스 상태는 변하지 않는다. 하지만 post는 동일한 내용을 반복 연산할 경우, 독립적인 resource가 따로 생겨난다.

Q : Server의 status가 뭔가????

Token과 Message ID

**Safe** : data의 value 바뀌는지 여부. put, post, delete를 제외한 get은 조회역할만 하므로 데이터를 변경하지 않는다.

**Asyncronous :**

동기식이 클라이언트와 서버간에 응답/요청이 바로 이루어지는 반면,

비동기식은 클라이언트의 요청을 받은 서버가 timeout을 초과할 경우, 나중에 해당되는 메시지를 보낸다. 즉 한번에 여러 클라이언트를 상대하기 위한 메커니즘.

CoAP 자체가 비동기적으로 사용된다. 즉, CoAP 네트워크 프로토콜을 사용하는 디바이스는 전원을 절약할 수 있으며, Sleep 모드로 대기해 메시지가 발생할 때만 일을 처리할 수 있다

**<IoT 구성 - Final>**

<**Home automation** >

집주인에게 알맞은 온도를 학습하여 귀가 시, 최적화된 온도 제공

비 올 때, 창문닫기