

조명 치료 서비스를 위한 IoT 시스템 설계

임종윤, 문재원, 정준호, 임태범
전자부품연구원 스마트 미디어연구센터
jyun1589@keti.re.kr, jwmoon@keti.re.kr, juunnee@keti.re.kr, tlim@keti.re.kr

IoT System Design for Light Therapy Service

Lim JongYun, Moon JaeWon, Jung JunHo, Lim TaeBeom

Korea Electronics Technology Institute Smart Media Research Center

요 약

본 논문은 실시간으로 변하는 사용자의 감정 상태에 적응하여 IoT 조명을 제어해 조명 치료 서비스를 제공하는 시스템을 제안한다. 이를 위해서는 사용자 및 조명의 현재 상태를 인지하고 색상 및 조도를 원하는 상태로 실시간 제어 가능해야 하기에 REST 방식의 API 를 설계하고 구현하였다. 제안하는 시스템은 기존의 조명들을 통합적으로 관리하기 어려웠던 문제를 해결하고 이를 기반으로 조명 치료에 적용하여 사용자에게 편리한 사용성과 조명을 다양한 방법으로 활용할 수 있는 확장성을 제공할 것이다. 제안하는 시스템은 업무공간, 호텔, 가정 등 다양한 장소에서 이용되고, 홈 IoT 디바이스 시장이 확장하고 있는 현시점과 맞물려 사용자들에게 충분한 이용가치를 지닌 시스템이 될 것으로 기대된다.

1. 서론

최근 들어 다양한 IoT 디바이스들이 출시되면서 네트워크 기반으로 디바이스를 관리할 수 있는 토대가 마련되었다[1]. 또한 이런 경향 때문에 IoT 분야와 플랫폼에 대한 연구가 활발히 진행되고 있다[2]. 이런 환경은 이전에 개별적으로 관리해야만 했던 디바이스들을 웹 또는 모바일 플랫폼 상에서 통합적으로 손쉽게 관리할 수 있도록 하였다. 본 시스템에서는 조명에 초점을 맞추어 특정 공간에 있는 IoT 조명기기를 통합적으로 관리 및 제어하고, 이를 조명 치료에 이용하여 빛을 이용한 조명 치료 서비스를 제공하는 시스템을 제안한다.

2. 조명 치료 서비스를 위한 시스템 설계

2.1 REST 기반 조명제어 API 설계

2000 년도에 Roy Thomas Fielding 이 REST 방식의 아키텍처를 제안하였다[3]. REST 방식은 stateless 한 특성 때문에 서버와 클라이언트 간의 동기화에 필요한 로드를 줄일 수 있어 효율적이다[4]. 또한 REST 방식을 사용할 경우 서버와 클라이언트의 역할이 확실히 구분되어 시스템간 의존도가 줄어들게 되는데, 이럴 경우 클라이언트 입장에서 서버 측의 API 만 호출하면 되는 이점이 있다.

본 논문에서는 조명 치료 서비스를 위한 시스템을 제안하고 조명을 제어하는 부분과 조명과 관련된 데이터를 주고받는 모듈을 REST 방식으로 구현하였다. REST 방식의 API 를 사용하여 조명 치료 서비스 시스템의 클라이언트와 서버 간 통신 구현이 명료해지고 제어 및 유지 보수가 간단해지는 장점을 갖게 된다.

2.2 시스템 구성도

그림 1 은 본 논문에서 제안하는 시스템의 구조를 나타낸 구성도이다.

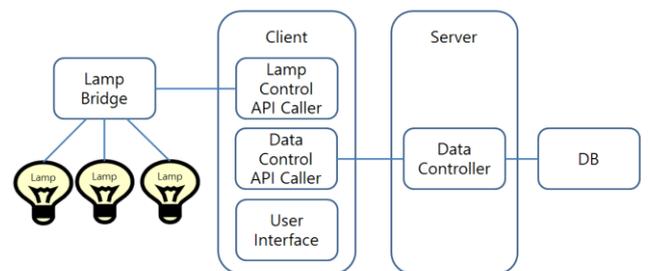


그림 1. 시스템 구성도

시스템은 크게 조명 및 브릿지와 직접 통신하는 클라이언트와 기존 사용 데이터 및 조명 설정 데이터를 저장하는 DB 를 조회 및 업데이트 하는 서버로 구성된다. 서버에는 DB 와 통신하며 데이터를 조회하거나 삽입 및 삭제하는 Data Controller 모듈이 있다. Client 측의 Data Control API Caller 는 Data Controller 에 구현되어 있는 API 를 REST 방식으로 호출하여 서버에서 DB 에 조명과 관련된 데이터를 조회하거나 삽입, 수정, 삭제하는 등의 작업을 수행할 수 있도록 한다. 실제로 가장 많은 데이터 이동이 일어나는 부분으로 DB 에는 조명 및 제어에 관한 정보가 기록되어 사용자의 이용 상태가 저장된다.

또한 클라이언트 측의 Lamp Control API Caller 는 조명 브릿지에 REST 방식으로 메시지를 전송하여 조명의 상태를 조회하거나 변경시키는 것이 가능하다. 클라이언트

측은 서버와 조명 브릿지 양방향에 REST 기반의 API 만 호출하면 되므로 구조가 간결하고 효율적인 통신이 이루어진다.

2.3 동작 시나리오

그림 2 는 서비스 시나리오에 따른 시스템 구성 요소의 동작을 나타낸 시퀀스 다이어그램이다.

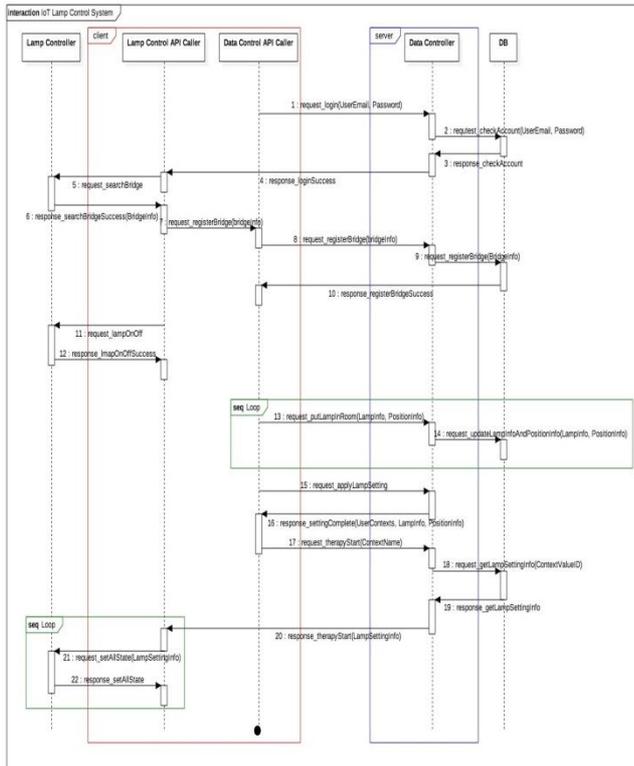


그림 2. 시나리오에 따른 시스템 내부 동작

사용자가 서비스에 진입하면 어플리케이션과 조명 사이의 매개체 역할을 하는 조명 브릿지를 탐색하는 과정이 시작된다. 동일한 네트워크 망에 있는 브릿지를 찾아 기기 정보를 조회하고 시스템 서버를 거쳐서 DB 에 브릿지의 정보를 저장하게 된다(1~10).

브릿지의 등록이 완료되면 사용자가 조명 아이콘을 선택하여 조명을 켜거나 끌 수 있다. 이 기능을 통해 각 조명 아이콘이 실제 어떤 조명에 연동되어 있는지를 파악할 수 있다. 이 제어 기능을 수행할 때에는 클라이언트 측의 Lamp Control API Caller 에서 Lamp Controller 의 API 를 호출하여 켜거나 끄는 요청을 전송한다(11~12).

사용자는 조명 아이콘을 Drag & Drop 방식으로 UI 상의 도면 위에 위치시킬 수 있다. 이 때, Drop 이벤트가 발생하면 그 조명 및 위치 정보를 Data Controller 에 REST 방식으로 전송하여 시스템 DB 에 저장한다. 조명 아이콘을 도면 상에 위치시키는 방식을 통하여 특정 위치에 있는 조명을 직관적으로 파악할 수 있다는 장점이 있다(13~14).

조명 관리 및 제어 기능뿐만 아니라 조명치료 서비스를 이용하고 싶다면 현재 감정을 선택하고 서비스를 시작하면 된다. 이 때에, 감정에 따라 다르게 선택된 조명의 색상군과 제어 알고리즘을 통해 조명의 제어가 시작된다. Lamp

Control API Caller 에서 Lamp Controller 에 주기적으로 제어 메시지를 전송하여 통신한다. 서비스가 종료되면 다시 해당 색상군의 초기 상태로 변경되어 정지한다(20~22).

2.4 시뮬레이션

본 시스템을 기반으로 2 개의 조명 브릿지와 18 개의 IoT 조명을 연동하여 테스트한 결과, 1 초 간격으로 모든 조명을 제어하여 5 분 이상의 조명 치료 서비스를 원활하게 제공하였다. 이를 통해 사용자의 심리 상태나 환경 상태에 기반하여 실시간으로 조명을 제어하고, 또한 사용자의 상황 정보에 적합한 치료 효과를 제공할 수 있었다. 이를 통해 본 시스템이 네트워크 통신 상에 문제가 생기지 않는다는 가정하에, 특정 공간에서 조명 제어 기능을 제공하고, 치료 서비스에 활용되는 데에 충분한 성능을 가지고 있음을 확인하였다.

3. 결론

본 IoT 조명 제어 시스템에 대하여 그 설계 및 구현 방식과 REST 기반으로 동작하는 과정을 알아보았다. IoT 조명이 출시되기 전까지 개인 사용자가 조명을 통합적으로 제어한다는 것을 생각하기 어려웠지만, IoT 조명과 본 시스템을 통하여 특정 공간에 있는 조명 그룹을 통합적으로 관리할 수 있다. 더불어 직관적인 인터페이스를 통해서 공간 내 조명의 위치에 대해서도 편리하게 파악이 가능하다. 또한 REST 방식을 기반으로 API 를 호출하는 시스템을 구성하여, 다양한 웹 서비스에서 통용될 수 있고 유지 보수 및 제어가 간편하게 이루어진다.

추후에 조명 제어 알고리즘을 추가하여 치료 뿐만 아니라 분위기에 맞는 다채로운 조명 효과를 낼 수 있는 서비스를 제공할 수 있을 것이다. 또한 조명 뿐만 아니라 도어락, 도어 개폐센서, 모션센서 등의 IoT 디바이스를 연동하여 다양한 IoT 디바이스들을 통합적으로 관리할 수 있는 시스템으로 확장할 수 있을 것이라 기대한다.

ACKNOWLEDGMENT

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터 (IITP)의 정보통신·방송 연구개발사업의 일환으로 수행하였음 [과제번호: R0101-16-0159, 기기 정보뿐 아니라 사용자의 환경/감성/인지 정보에 적응적으로 반응하는 정보 기기용 원격 UI 기술 개발]

참고문헌

[1] Gubbi, Jayavardhana, et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future Generation Computer Systems* 29.7 (2013): 1645-1660.

[2] 홍경완, and 신동군. "사물 인터넷 분야의 연구 및 표준화, 플랫폼 동향." *텔코 저널* 3.단일호(2015):169-190.

[3] Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)"

[4] Alex, Rodriguez(2008). "RESTful Web services: The basics." IBM developer Works.