

# 웹 애플리케이션 성능 분석을 위한 REST API 기반의 서버 구축

## REST API based Server Construction for Web Application Performance Analysis

김현국\* · 박진태 · 최문혁 · 문일영  
한국기술교육대학교 컴퓨터공학부

Hyun-gook Kim\* · Jin-tae Park · Moon-Hyuk Choi · Il-young Moon

Department of Computer Engineering, Korea University of Technology and Education, Chungcheongnam-do, 31253, Korea

### [요 약]

4차 산업혁명을 통해 웹이 다양한 기술들과 융합되기 전에는 웹 애플리케이션의 성능을 평가하는 요소가 단순히 웹 애플리케이션이 로딩 되는 속도뿐이었다. 따라서 기존에 시중에 나와 있는 웹 애플리케이션 성능분석 툴의 경우 대부분 웹 애플리케이션이 웹 브라우저에서 동작 하는 속도에 초점을 맞추고 있다. 하지만 웹은 이제 단순 웹 브라우저 뿐만 아니라 다양한 서비스들과 융합되어 활용되고 있으며, 단순 속도 이외에도 웹 애플리케이션을 평가하기 위해 평가해야할 요소들이 등장했다. 따라서 본 논문에서는 앞서 언급한 속도뿐만 아니라 프로그레시브 웹 앱, 접근성, 모범 사례, 검색 엔진 최적화를 고려한 웹 애플리케이션 성능 분석을 진행하기 위한 서버를 설계하고, REST API 형태로 서비스를 구축하여 사용자의 기기나 환경에 영향을 받지 않고 웹 애플리케이션 성능 서비스를 제공할 수 있는 서버를 설계 및 구현하고자 한다.

### [Abstract]

Until the 4th Industrial Revolution fused the Web with diverse technologies, the only factor in evaluating the performance of a Web application was the loading speed of the Web application. Therefore, most existing Web application performance analysis tools focus on the speed at which Web applications run in a Web browser. However, the web is now being used not only in a simple web browser but also in a variety of services. So, in addition to simple speed, there are more elements to be checked to evaluate web applications. Therefore, in this paper, we design a server to perform web application performance analysis considering progressive web app, accessibility, best practice, and search engine optimization as well as the above mentioned speed. Also, we realize a REST API based server that can provide web application performance services without being affected by user's devices or environment.

**Key word** : Web performance, PWA, Representational state transfer, Performance analysis, API.

<https://doi.org/10.12673/jant.2018.22.5.456>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 28 September 2018; Revised 1 October 2018  
Accepted (Publication) 23 October 2018 (30 October 2018)

\*Corresponding Author; HyunGook Kim

Tel: +82-10-5025-0874

E-mail: hy1392@koreatech.ac.kr

## 1. 서론

4차 산업혁명을 통해 웹이 다양한 기술들과 융합되기 전에는 웹 애플리케이션의 성능을 평가하는 요소가 단순히 웹 애플리케이션이 로딩 되는 속도뿐이었다. 그 당시의 웹의 기능은 단순 정보전달을 목적으로 하고 있었으며, 따라서 웹 애플리케이션을 이용하는 사용자들이 정보를 신속하고, 정확하게 제공하는 것이 가장 중요한 요소였다. 그 결과 기존에 시중에 나와 있는 웹 애플리케이션 성능분석 툴의 경우 대부분 웹 애플리케이션의 동작 속도에 초점을 맞추고 있다. 하지만 웹이 점차 다양한 기술들과 융합되어 정보 전달 이외에 다양한 기능을 제공하기 시작하면서 단순 속도 이외에도 웹 애플리케이션을 평가하기 위해 평가해야할 요소들이 등장했다 [1].

가장 대표적인 예로는 프로그레시브 웹 앱(PWA; progressive web app)을 들 수 있다. 프로그레시브 웹 앱은 구글 크롬 엔지니어 알렉스 러셀이 2015년도에 고안해낸 기술로 앱의 수준으로 점진적으로 발전해 나가는 웹을 지향하는 것을 의미한다. 그림 1에서 나타나는 특징처럼 프로그레시브 웹 앱이 궁극적으로 추구하는 것은 일반적인 스마트폰의 앱 수준과 같은 사용자 경험을 웹을 통해 사용자에게 제공하는 것이다. 따라서 이를 지원하기 위해서는 웹 애플리케이션이 오프라인 상태 서비스 지원을 위한 서비스 워커(service worker)를 포함해야 하며, 캐싱을



그림 1. 프로그레시브 웹 앱의 특징  
Fig. 1. Features of Progressive web app.

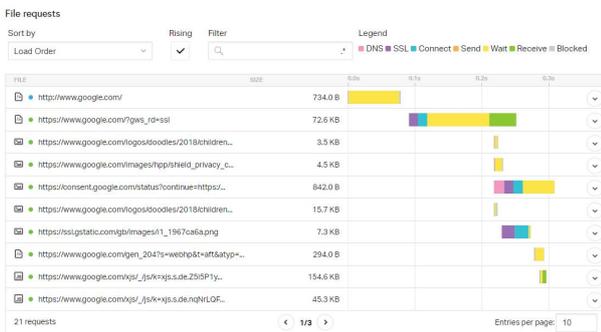


그림 2. Pingdom.com에서 제공하는 웹 애플리케이션 성능 분석 결과  
Fig. 2. Web Performance Analysis Result from Pingdom.com.

통해 파일을 제공할 수 있어야 한다. 또한 웹 앱 매니페스트(web app manifest) 파일을 가지고 있어 사용자가 웹 애플리케이션을 마치 네이티브 애플리케이션처럼 설치할 수 있어야 한다. 뿐만 아니라 HTTPS를 통한 보안성 및 크로스 브라우징(cross browsing)까지 지원할 수 있어야 한다.

앞서 설명한 프로그레시브 웹 앱 이외에도 추가로 평가해야 할 사항들이 있는데, 웹을 이용하는데 불편함이 있는 사용자들을 돕기 위한 접근성(accessibility), 웹 애플리케이션이 항상 최신 웹 표준 및 사양을 준수하기 위한 모범 사례(best practices) 그리고 웹 정확성 및 신뢰도 향상을 위한 검색 엔진 최적화(search engine optimization)이다. 따라서 오늘날에 웹 애플리케이션을 분석한다고 하면 앞서 언급한 모든 요소들을 아우를 수 있는 평가가 진행되어야 한다 [2],[3].

하지만 기존에 웹을 통해 제공되고 있는 웹 애플리케이션 분석 서비스나 설치 형태로 제공되는 웹 애플리케이션 분석 서비스의 경우 그림 2에 나타나는 것처럼 단순 파일들의 로딩 속도를 측정하는 데에 머무르고 있다. 따라서 속도 이외에도 중요하게 평가되는 항목들에 대한 분석을 진행하는 데에 어려움이 있다. 또한 이러한 서비스들은 웹 사이트나 설치 가능한 프로그램의 형태로 배포가 진행되기 때문에 웹 브라우저를 통해 직접 해당 서비스를 접근하거나, 서비스 설치 프로그램을 다운로드 받아 직접 설치하여 성능 분석을 진행해야 한다 [4],[5]. 이는 해당 분야에 전문지식을 가지고 있지 못해 코딩에 어려움이 있는 사람들에게 큰 방해요소로 작용하게 된다. 따라서 본 논문에서는 앞서 언급한 속도뿐만 아니라 프로그레시브 웹 앱, 접근성, 모범 사례, 검색 엔진 최적화를 고려한 웹 애플리케이션 성능 분석을 진행하기 위한 서버를 설계하고, REST API 형태로 구현하여 웹 사이트에 접속하거나, 설치를 진행할 필요 없이 모든 기기에서 HTTP만을 이용하여 성능 분석을 진행하고, 원하는 항목에 대한 분석 결과 값만을 선택적으로 받아와 활용할 수 있는 서비스를 구축하고자 한다.

## II. 관련 기술 조사

연구를 통해 본 논문에 담고자 하는 서버는 몇 가지 특징을 가지게 된다. 먼저 다양한 플랫폼을 지원하기 위해 REST API 기반으로 구축된다. 웹 서버가 사용자의 요청에 대해 데이터를 처리하고, 처리한 데이터를 기반으로 결과 화면을 구성해 HTML 파일 자체를 반환하지 않는다. 웹 서버는 사용자의 요청을 처리하여 분석 결과 값만을 반환하여 준다. 따라서 사용자는 서버에 대한 요청으로부터 받은 결과 값을 기반으로 각자 데이터를 활용하여 필요한 화면을 구성하게 된다. 다음은 본 논문에서 진행한 연구의 바탕이 되는 기술들에 대한 설명이다.

REST(representational state transfer)는 월드 와이드 웹과 같은 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식으로 2000년 로이 필딩(R. Fielding)의 박사학위 논문에서

서 처음으로 소개된 개념이다. 로이 필딩은 HTTP의 주요 저자 중 한 사람으로 현재 이 기술적 개념은 네트워킹 문화에 널리 사용되고 있다. 한 마디로 정의하면 REST는 네트워크를 통해 공유되는 각각의 자원을 정의하고 자원에 대한 고유 주소를 지정하는 방법 전반을 일컫는다 [6].

REST는 몇 가지 특징을 가지고 있는데 각각의 특징은 다음과 같다.

1) uniform interface: URI로 지정한 리소스에 대한 조작을 통일되고 한정적인 인터페이스로 수행하는 아키텍처를 갖추어야 한다.

2) stateless : 작업을 위한 상태정보를 따로 저장하고 관리하지 않는다. 세션이나 쿠키를 사용하여 별도로 정보들을 저장하고 관리하지 않기 때문에 API 서버는 사용자로부터 들어오는 요청만을 단순히 처리한다. 따라서 서비스의 자유도가 높아지고 서버에서 불필요한 정보를 관리하지 않음으로써 구현이 단순하다.

3) cacheable : HTTP라는 기존 웹 표준을 그대로 사용하기 때문에, 웹에서 사용하는 기존 기술들을 그대로 활용이 가능하다. 따라서 HTTP가 가진 캐싱 기능이 적용 가능하며, HTTP 프로토콜 표준에서 사용하는 Last-Modified태그나 E-Tag를 이용하면 캐싱 구현이 가능하다. 이를 통해 확장성과 성능을 향상시킬 수 있다.

4) self-descriptiveness : REST API 메시지만 보고도 이를 쉽게 이해 할 수 있는 자체 표현 구조 구성된다.

5) client - server 구조 : REST 서버는 API 제공, 클라이언트는 사용자 인증이나 컨텍스트(세션, 로그인 정보)등을 직접 관리하는 구조로 각각의 역할이 확실히 구분된다. 따라서 클라이언트와 서버간의 역할이 분명하며, 서로간의 의존성이 약해진다.

6) 계층형 구조 : REST 서버는 다중 계층으로 구성될 수 있으며 보안, 로드 밸런싱, 암호화 계층을 추가해 구조상의 유연성을 둘 수 있고 proxy, 게이트웨이 같은 네트워크 기반의 중간 매체를 사용할 수 있다 [7].

REST의 특징에서도 확인할 수 있듯이 REST 기반의 API 서버는 HTTP를 기반으로 웹뿐만 아니라 다양한 모바일 기기 및 전자 기기에서 접근이 가능하기 때문에 분석 결과를 다양한 장치에서 활용할 수 있다는 장점이 있다. 따라서 본 논문에서는 REST API를 기반으로 하여 서버를 구축하였으며, 다양한 기기에서 HTTP 통신만 가능하다면 성능 분석을 진행하고 결과 값을 활용할 수 있도록 설계하였다.

### III. REST API 기반의 웹 애플리케이션 성능 분석 서버 설계

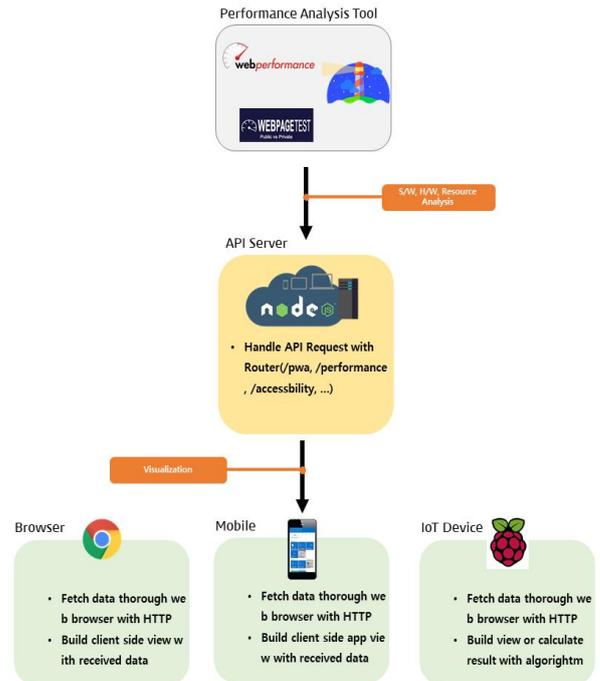


그림 3. 웹 애플리케이션 성능 분석을 위한 API 서버 동작도  
 Fig. 3. Operation diagram of API Server for web application performance analysis.

표 1. API URL별 역할 구성

Table 1. Configured roles by API URL.

index	API URL	Role
1	/Performance	Return performance analysis result of received URL
2	/PWA	Return PWA analysis result of received URL
3	/Accessibility	Return accessibility analysis result of received URL
4	/BestPractice	Return best practice analysis result of received URL
5	/SEO	Return search engine optimization analysis result of received URL
6	/Analysis	Return all analysis result of received URL

그림 4. API 서버에서 반환되는 JSON 결과 값 예시  
 Fig. 4. Example of JOSN result from API server.

```
{
  "Main Category": "SEO",
  "Main Score": "75",
  "Analysis List": [
    {
      "Analysis Title": "SEO Part 1",
      "Weight": 1,
      "Sub Category": "seo-mobile",
      "Result": {
        "spent time": 1.35874,
        "score": 50,
      },
    },
    ...
  ]
}
```

본 논문에서 제안하고자 하는 웹 애플리케이션 성능 분석 API 서버는 그림 3과 같은 구성을 가진다.

먼저, 서버 구축에는 그림의 API server 부분과 같이 node.js 를 기반으로 구축되었다. node.js는 V8 자바스크립트 엔진을 기반으로 동작하는 오픈소스로 개발 생산성, 코드의 양을 줄이는 JSON지원, 비동기 프로그래밍 지원, ECMA Script의 적용, 성능 및 안정성, 광대한 커뮤니티 등 다양한 장점을 가지고 있는 자바스크립트 런타임 환경이다. 하지만 무엇보다도 중요한 것은 하드웨어 확장성이 뛰어나다는 것이다. 4차 산업혁명을 통해 등장한 사물 인터넷과 같은 기술에 적용되기 위해서는 제약된 하드웨어 인프라에서 확장성이 좋아야 한다. 따라서 JSON을 기반으로 의존 패키지들을 관리하고, 손쉽게 설치 및 이식이 가능케 지원하는 node.js가 가장 적합한 서버 환경이라 생각하여 이를 기반으로 구축하였다. 서버 내에서는 HTTP통신을 진행하기 위해 node.js의 웹 프레임워크인 express를 이용해 서비스를 구동하였고, 각각의 URL에 진행되어야할 성능 분석을 매핑시켜 환경을 구성하였다. 실제로 구성된 환경의 API URL과 각 URL의 역할은 다음 표 1과 같다. 각각 이름에 해당하는 결과 값을 JSON의 형태로 사용자에게 반환하여주고 있으며, 공통적으로 분석을 진행하기 위해 필요한 웹 사이트의 URL을 인자 값으로 받아와 분석을 진행하는 데에 사용하고 있다.

반환하는 데이터의 형태는 동일하게 JSON 형태이며, 사용자가 사용하기 쉽게 분석 항목별로 그룹으로 묶여 제공된다. 따라서 사용자는 해당 항목의 전체 점수를 사용할 수도 있고, 필요에 따라 원하는 항목의 점수만을 추출하여 사용할 수 있도록 제공한다. 실제 API 서버를 통해 반환되는 JSON 결과 값의 형태는 다음 그림 4와 같다. 그림 4는 SEO에 대한 분석 결과 값을 요청하였을 때의 예시이다.

예시에서 볼 수 있듯이 각각의 대 분류별로 하나의 결과 항목이 생성된다. 실제 SEO라는 검색 엔진 최적화 대분류 안에 Analysis List라는 세부 분석 항목들이 포함되어 있으며, 이 안

에 실제 진행된 분석 항목들이 담기게 된다. 각각의 분석 항목들은 분석 항목을 설명하는 분석 명, 분석에서 중요도를 나타내는 Weight, 소분류를 나타내는 Sub Category, 그리고 실제 수행 결과 값을 나타내는 Result로 구성된다. 실제 수행 결과 값을 나타내는 Result 안에는 실제 해당 항목을 분석하는 데에 소요된 시간을 나타내는 spent time, 소요 시간을 기반으로 산출된 점수를 나타내는 score 항목이 포함되어 있다. 대분류에 포함된 대표 점수인 Main Score는 모든 분석 항목들에 대한 환산 점수 score를 평균 낸 것이다.

#### IV. REST API 기반의 웹 애플리케이션 성능 분석 서버 구현

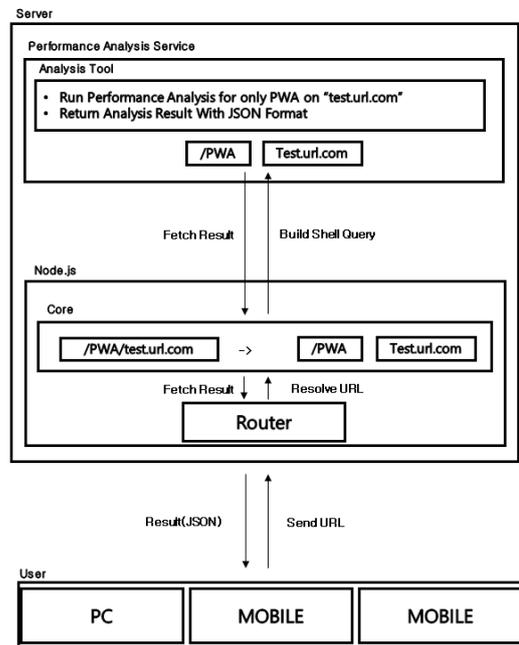


그림 5. 웹 애플리케이션 성능 분석을 위한 API 서버 동작 예시  
 Fig. 5. Example of API server behavior for Web application performance analysis.

그림 6. 전체 분석에 대한 API 구축 예시  
 Fig. 6. Example of building an API for full analysis.

```
app.post('/Analysis', (req, res) => {
  if (shell.exec("MyAnalysis "+ req.body.url + " --output-path=./results/report.json --output json --chrome-flags='--headless'")) {
    fs.readFile("./results/report.json", function(err, data){
      return res.json(JSON.parse(data))
    })
  }
  else {
    return res.json("error")
  }
});
```

상기 그림 6에 명시된 자바스크립트 코드는 실제 node.js를 이용하여 API 서버를 구축하는데 사용된 코드의 예시이다. node.js의 라우터를 이용하여 /Analysis, /Performance, /BestPractice와 같이 사용자로부터 입력받은 URL을 기반으로 각각에 해당하는 분석만을 실행할 수 있도록 분기처리를 진행하였다. 이후 각각의 분석 요청에 대하여 성능 분석을 처리하게 된다. 그림 6의 코드는 그림 5의 Node.js 부분을 나타낸 것이라 볼 수 있다.

전체적인 동작의 프로세스는 그림 4에서 나타나는 것과 같다. pc, mobile 및 다양한 IoT 기기들로부터 HTTP 통신을 이용하여 분석하고자 하는 항목과 사이트에 대한 정보를 넘겨받는다. 사용자는 HTTP가 지원되는 환경이라면 어떤 기기 및 환경에서라도 서비스를 이용할 수 있으며, 간단히 분석하고자 하는 정보와 해당하는 URL만을 API 서버로 전달하여주면 된다. 전달된 URL은 node.js의 라우터를 거쳐 해당하는 API URL과의 매칭, 분석 대상이 되는 URL로 분류되어 실제 성능 분석을 진행하는 데에 사용될 shell 쿼리를 작성하는데 사용된다. 예시 그림에서는 PWA 항목에 대한 분석을 사용자가 요청하였으며, 요청하고자 하는 URL 주소는 "test.url.com"이다. 따라서 이 정보를 바탕으로 API 서버에서 작성된 shell 쿼리가 실제 분석 틀에 전송되고, 해당 URL에 대한 성능 분석이 진행된다. 분석 결과는 JSON의 형태로 서버에서 가공되어 그림 4와 같은 형태로 다시 사용자들에게 전달된다. 따라서 사용자들은 그림 4의 형태로 전달된 데이터를 바탕으로 성능 분석 결과를 디버깅하거나, 데이터를 바탕으로 한 새로운 서비스를 창출해낼 수 있다.

## V. 결론

본 논문에서는 기존에 사용되고 있는 웹이나 설치를 통해 사용자가 사용하는 웹 애플리케이션 성능 분석 툴들의 한계점에 대하여 논하고, 접근성이라는 가장 큰 장벽을 허물기 위해 REST 기반의 API 서버를 설계 및 구현하였다. 논문에서 제안한 방법을 통하여 그동안 웹 애플리케이션 성능 분석을 위해서는 특정 하드웨어에 종속되어 프로그램을 작성하거나 사용해야 했던 것과 달리 HTTP 통신이 가능한 모든 기기에서 접근이 용이해져 다양한 기기가 분석을 진행하고, 결과 값을 활용해 새로운 결과를 창출해낼 수 있는 방법을 제안하였다. 현재 논문에서는 서론에서 제안한 5가지 항목에 대한 성능 분석 결과만을 제공하고 있지만, 연구를 통해 제공하는 항목을 점차 확장하고

제공되는 데이터의 형태 또한 넓힐 계획이다. 따라서 다음 논문에서는 본 논문에서 제안한 논문의 시스템을 확장하여 더 많은 분석 항목들을 JSON 이외의 다른 형태로 사용자에게 제공할 수 있는 서비스를 제안하고자 한다.

## Acknowledgments

본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 사회맞춤형 산학협력 선도대학(LINC+) 육성사업의 연구결과입니다.

## References

- [1] V. Atlidakis, P. Godefroid and M. Polishchuk, "REST-ler: automatic iIntelligent REST API Fuzzing", arXiv preprint arXiv:1806.09739, pp. 1-14, 2018.
- [2] W. Fu, T. D'Ottavio and S. Nemesure, "A generic REST API service for control databases", in *16th International Conference on Accelerator and Large Experimental Physics Control Systems*, Barcelona; Spain, pp. 465-468, 2018.
- [3] R. T. Fielding and R. N Taylor, *Architectural styles and the design of network-based software architectures*, 7th ed, Doctoral dissertation: University of California, Irvine, 2000.
- [4] S. P. Ong, S. Cholia, A. Jain, M. Brafman and D. Gunter, "The materials application programming interface (API): A simple, flexible and efficient API for materials data based on representational state transfer (REST) principles," *Computational Materials Science* 97, Vol. 1, pp. 209-215, 2015.
- [5] J. Zou, J. Mei and Y. Wang, "From representational state transfer to accountable state transfer architecture," in *2010 IEEE International Conference on Web Services*, Miami; FL, pp. 299-306, 2010.
- [6] R. Secchi, A. C. Mohideen and G. Fairhurst, "Performance analysis of next generation web access via satellite", *International Journal of Satellite Communications and Networking*, Vol. 36, No. 1, pp. 29-43, 2018.
- [7] [Internet]. Available: <https://meetup.toast.com/posts/92>



**김 현 국 (Hyun-Gook Kim)**

2017년 8월 ~ 현재: 한국기술교육대학교 컴퓨터공학과 석사과정  
※관심분야: 사물인터넷, 웹 어셈블리, 웹 표준



**박 진 태 (Jin-Tae Park)**

2015년 8월 : 한국기술교육대학교 컴퓨터공학과 (공학석사)  
2015년 9월 ~ 현재 : 한국기술교육대학교 컴퓨터공학과 박사과정  
※관심분야 : Web Assembly, Standardization of Web Technologies, Web Application Engineering



**최 문 혁 (Moon-Hyuk Choi)**

2014년 2월 ~ 현재: 한국기술교육대학교 컴퓨터공학과 학사과정  
※관심분야: 인공지능, IoT



**문 일 영 (Il-Young Moon)**

2000년 2월 : 한국항공대학교 항공통신정보공학과 졸업 (공학사)  
2002년 2월: 한국항공대학교 대학원 항공통신정보공학부 졸업 (공학석사)  
2005년 2월: 한국항공대학교 대학원 정보통신공학과 졸업 (공학박사)  
2004년 ~ 2005년 : 한국정보문화진흥원 선임연구원  
2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수  
※관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP