

## 조선해양기자재 제작을 위한 SCM 기반 실시간 데이터 인터페이스 기술에 관한 연구

A Study on the Real-time Data Interface Technology based on SCM for Shipbuilding and Marine Equipment Production

한명기<sup>1</sup>, 김영훈<sup>2\*</sup>, 박준수<sup>3</sup>, 이원호<sup>4</sup>

Myeong-Ki Han<sup>1</sup>, Young-Hun Kim<sup>2\*</sup>, Jun-Su Park<sup>3</sup>, Won-Ho Lee<sup>4</sup>

### 〈Abstract〉

The production and procurement of shipbuilding and offshore equipment is an important competitive factor in the shipbuilding and offshore industry. Recently, ICT-based digital technology has been rapidly applied to the manufacturing industry following the Fourth Industrial Revolution. Under the digital transformation, real-time data interface technology based on SCM (Supply Chain Management) is emerging as an important tool to improve the efficiency of the equipment manufacturing process. In this study, the characteristics and advantages and disadvantages of interface technologies of web-based data interface technologies were compared and analyzed. The performance was compared between theoretical evaluation based on technical features and practical application cases. As a result, it was confirmed that GraphQL is useful for selective data processing, but there is a problem with optimization, and REST API has a problem with receiving data due to a fixed data structure. Therefore, this study aims to suggest ways to utilize and optimize these data interface technologies.

*Keywords : Supply Chain Management (SCM), Real-Time Data, OAuth2 Authentication, GraphQL, REST API, Data Management Techniques, Web Security*

1 주저자, 경남대학교, 메카트로닉스공학과, 석사과정  
E-mail: mkhan@hanwha.com

2\* 정회원, 교신저자, 경남대학교, 조선해양시스템공학과, 교수  
E-mail: younghun@kyungnam.ac.kr

3 경남대학교 조선해양시스템공학과, 교수  
E-mail: junsoopark@kyungnam.ac.kr

4 경남대학교 메카트로닉스공학과, 석사과정  
E-mail: m\_yu@daum.net

1 Main Author, Dept. of Mechatronics Engineering, Kyungnam University, Graduate student

2\* Corresponding Author, Dept. of Naval Architecture & Ocean System Engineering, Kyungnam University, Professor

3 Dept. of Naval Architecture & Ocean System Engineering, Kyungnam University, Professor

4 Dept. of Mechatronics Engineering, Kyungnam University, Graduate student

## 1. 서론

조선·해양 산업은 우리나라의 주력산업의 하나로 국민경제적으로 큰 역할을 하고 있다. 이러한 산업의 핵심에는 기술 혁신 및 연구 개발 활동이 지속적으로 이루어져 왔으며, 이를 통해 높은 국제 경쟁력을 유지하고 있다. 그 중에서도 조선해양기자재 제작 및 조달은 산업 전반의 품질과 생산성, 그리고 경쟁력 향상에 결정적인 역할을 하는 과정으로 매우 중요하다. 특히 최근 디지털 변혁의 환경 하에서 SCM(Supply Chain Management) 기반의 실시간 데이터 인터페이스 기술은 기자재 제작 프로세스의 효율성을 크게 향상시킬 수 있는 중요한 도구로 부상하였다.

2000년대 초반 첫 번째 실시간 데이터 인터페이스 기술 연구는 주로 ERP시스템의 데이터 통합과 관련된 주제에 초점을 맞추었다. 이 시기의 연구들은 기본적인 데이터 통합 방법론과 프로토콜에 대한 논의가 주를 이루었다. 2010년대 들어서는 GraphQL 및 REST API 같은 새로운 데이터 인터페이스 기술의 등장으로 기존의 연구 패러다임이 크게 바뀌었다. 이러한 기술들은 데이터의 선택적 요청 및 응답을 가능하게 하여 SCM의 효율성을 크게 향상시키는데 기여하였다. 최근 연구들은 다양한 데이터 인터페이스 기술들을 통합하거나 최적화하는 방안에 관심을 가지고 있다. 특히, 데이터 초과 수신(over-fetching) 및 데이터 미달 수신(under-fetching) 문제를 해결하기 위한 다양한 알고리즘 및 방법론에 대한 연구가 활발히 진행되고 있다. 조선해양기자재 제작을 위한 SCM 기반 실시간 데이터 인터페이스 기술은 지속적인 발전을 거듭하고 있으며, 향후에도 더욱 고도화된 데이터 인터페이스 기술의 등장으로 SCM 시스템의 효율성 향상을 위한 경쟁은 치열해질 것으로 보인다.

그러나, 실제 국내 산업 현장에서는 다양한 데이터 인터페이스 기술 중 어떠한 기술을 선택하고, 어떻게 적용해야 최적의 결과를 가져올 수 있는지에 대한 명확한 가이드라인이 부족하다. 이에 따라 많은 기업이 불필요한 시간 및 비용을 낭비하며, 기술의 잠재력을 제대로 활용하지 못하는 문제가 지속적으로 발생하고 있다. 또한 조선·해양 산업의 핵심 기술 및 데이터가 경쟁국에 유출되지 않도록 보호하는 기술보안과 대·중·소 상생협력 측면의 효율성도 유지하는데 매우 중요한 경쟁요소가 되고 있다.

본 연구는 이러한 문제점을 해결하고자, 조선해양기자재 제작을 위한 SCM 기반 실시간 데이터 인터페이스 기술의 현재 상황 및 관련 기술의 비교 분석을 통해 좀 더 효율적인 기술 및 적용 방안을 제안하고자 한다.

## 2. 인터페이스 기술

### 2.1 인터페이스 기술 변화

인터페이스 기술은 서로 다른 두 시스템이나 컴포넌트가 정보를 서로 주고받을 수 있도록 하는 통신 방법이나 규약을 의미한다. 인터페이스 기술의 종류로는 파일기반 인터페이스, 데이터베이스 링크, 웹 서비스 등이 있다. 2010년대 들어서는 GraphQL 및 REST API 같은 새로운 데이터 인터페이스 기술의 등장으로, 기존의 연구 패러다임이 크게 바뀌었다. 이러한 기술이 각광받는 이유는 보안성이 편리성 보다 우선되어야 하기 때문이다. GraphQL과 REST API는 웹 기반 인터페이스 기술로 각각 다른 특성이 있다. GraphQL은 클라이언트가 필요한 데이터를 정확하게 요청할 수 있도록 하는 유연성을 제공한다. REST API는 산업계

에 널리 사용되는 표준 프로토콜을 기반으로 하며, 간단하고 이해하기 쉬운 구조로 되어 있다. 이 두 기술만 비교하는 이유는 최신 웹 개발에서 가장 주목받고 있는 인터페이스 기술이기 때문이다. REST API는 이미 널리 사용되고 있으며, GraphQL은 최근 발전하고 있는 새로운 기술로, 데이터 효율성과 통신 최적화에서 그 장점이 대두되고 있기 때문이다. 이러한 기술들은 데이터의 선택적 요청 및 응답을 가능하게 하여, SCM의 효율성을 크게 향상하는데 기여할 예정이다. 따라서 API에 의한 데이터 접근 및 통합은 SCM에서 매우 중요한 주제로 부상하고 있다. 본 논문에서는 파일기반 인터페이스, 데이터베이스 링크 등 보안문제가 존재하는 인터페이스 방식은 제외하고 산업에서 대표적으로 사용하고 있는 Web 기반으로 사용되고 있는 REST API와 GraphQL의 각각의 특징과 그 차이점을 살펴보고자 한다.

## 2.2 주요 데이터 인터페이스 기술들

GraphQL은 단일 엔드포인트를 사용하여 클라이언트가 필요한 정확한 데이터를 요청할 수 있게 하는 쿼리 언어이자 런타임이다. 클라이언트는 하나의 API 엔드포인트를 통해 구조화된 쿼리를 보내고, 서버는 해당 쿼리에 정확히 일치하는 데이터만을 응답으로 반환한다. 이는 클라이언트가 여러 엔드포인트를 호출할 필요 없이, 하나의 요청으로 복잡한 데이터 요구 사항을 충족시킬 수 있게 해준다. 예를 들어, 클라이언트는 게시물(posts), 댓글(comments), 작성자(authors) 정보를 하나의 쿼리로 요청하고, 서버는 이 세 가지 요구를 한 번에 처리하여 필요한 데이터만을 선택적으로 제공할 수 있다. 이 방식은 오버페칭과 언더페칭을 방지하고, 네트워크 효율성을 증가시킬 수 있는 장점이 있다.

REST API는 각각의 자원에 대해 별도의 엔드포인트를 가진다. 클라이언트는 이러한 엔드포인트를 통해 필요한 데이터를 요청하게 된다. Fig. 1에서는 “/posts”, “/comments”, “/authors” 등의 엔드포인트를 통해 각각의 자원에 접근하는 것을 보여준다. REST API는 각각의 자원에 대해 별도의 엔드포인트를 가진다.

두 API의 주요 차이점은 데이터 요청 및 수신 방식에 있다. GraphQL은 보다 유연한 데이터 요청이 가능하나, 복잡한 쿼리나 최적화 문제가 발생할 수 있다. 반면, REST는 구조화된 방식으로 데이터를 요청하며, 과도한 데이터 전송이나 under-fetching 문제가 있을 수 있다. 두 API 스타일 모두 장단점이 있으며, 특정 애플리케이션 또는 요구 사항에 따라 적합한 방식을 선택하는 것이 중요하다.

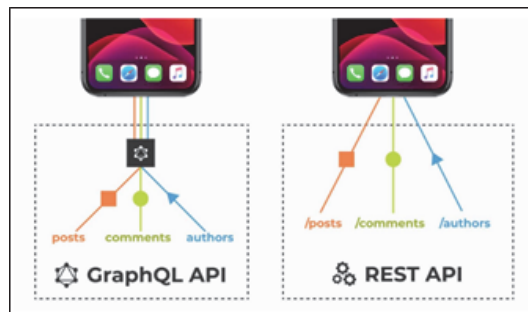


Fig. 1 Interface Method GraphQL and REST API

## 3. 데이터 인터페이스 방법의 성능비교

조선 및 해양 산업에서의 데이터 관리 및 통신은 다양한 기술들을 통해 이루어지고 있다. Excel은 기본적인 데이터 관리 도구로 널리 채택되어 있으며, 데이터의 다운로드 및 업로드에 주로 사용된다. 특히 단순한 데이터 관리 및 통합 작업에

있어서 매우 유용하다.

그러나 복잡한 데이터 구조와 통신 요구사항의 경우, GraphQL과 REST API 같은 고급 데이터 인터페이스 기술들을 활용해야 한다. GraphQL은 클라이언트 중심의 데이터 요청 방식을 제공하며, 클라이언트에서 필요한 데이터만을 선택적으로 요청할 수 있는 특징을 가지고 있다. 이는 특정 API, 예를 들면 GET 요청을 통해 다양한 데이터를 유연하게 접근할 수 있도록 해준다. 그러나 이 기술에는 DATA 송/수신 과정에서의 최적화 문제점이 존재한다.

반면에 REST API는 널리 알려진 HTTP 프로토콜을 기반으로 데이터를 요청하고 응답받는다. 이 방식의 장점은 국제 표준 프로토콜을 기반으로 한 일관성 있는 데이터 통신 방식을 제공한다는 것이다. 그러나 REST API를 사용할 때도 데이터의 초과 수신(over-fetching) 및 데이터 미달 수신(under-fetching) 같은 문제점이 발생할 수 있다. 이러한 문제점들은 실제 산업 현장에서 데이터 관리 및 통신의 효율성을 저하시키는 주요 요인으로 작용한다.

GraphQL과 RestAPI 정량적 분석으로 페이지 처리시간을 비교해 보았다. 페이지 처리시간의 핵

심 성능 지표로는, 사용자의 요청에 대한 서버로부터 첫 바이트를 수신하는데 걸리는 시간지표의 TTFB(Time to First Byte)와 전체 Content Download 시간으로 비교해 볼 수 있다.

Fig. 2와 같이 성능 테스트에 사용한 데이터는

GraphQL			
M_CODE	I_CODE	MATNR	CABL_TYPE
1001-0001	1076531136		
1001-0002	1076519184		
1001-0003	1076447197		
1001-0004	1076519185		
1001-0005	1076600811		

REST API			
M_CODE	I_CODE	MATNR	CABL_TYPE
1001-0006	1076576254	ECLALRG06US	RG6U-S
1002-0001	1076697936	4366MECVBECA0210	RPS2
1002-0002	1076855197	ECLA00S10UY	RG10UY-S
1002-0003	1076855198	ECLALRG06US	RG6U-S
1003-0001	1076878506	ECLALRG06US	RG6U-S

Fig. 2 Result of Test Program

Table 1. Technical evaluation by interface method

interface method	Advantages	Disadvantages
Excel File	-Familiar and user friendly -Widely accessible	-Limited scalability -Not suitable for complex data handling
GraphQL	-Selective data requests -Efficient data loading	-Complexity in query optimization -Steeper learning curve
RESR API	-Standardized and well understood -Good for fixed data sets	-Can lead to data over-fetching or under-fetching

Table 2. Page processing time comparison

Items	GraphQL (A)	REST API (B)	Speed Results
Columns	1	100	
Rows	100,000	100,000	
TTFB	1,380ms	159.77ms	REST API is 8.6x faster
Content Download	45.72ms	353.42ms	Graph QL is 7.7x faster
Total Time	1430ms	513.69ms	REST API is 2.8x faster

100,000건의 용량으로 비교하였다. 그 결과는 Table 2과 같이 GraphQL과 Rest API의 페이지 처리시간 결과 및 분석 결과를 얻게 되었다.

Table 2에서 보는 바와 같이 TTFB측면에서는 GraphQL 보다 REST API가 8.6배 빠른 반면 전체 Content Download시간은 Graph QL이 7.7배 빠른 결과를 가져왔다. 최종 Total Time은 REST API가 2.8배 빠른 결과를 가져와 결국 처리시간 측면에서는 REST API가 우수한 성능을 내었다.

GraphQL은 클라이언트 측에서 필요한 데이터 필드만을 명시적으로 요청할 수 있는 기능을 제공한다. 이는 데이터의 오버페칭(over-fetching)과 언더페칭(under-fetching)을 방지하며, 결과적으로 네트워크 부하를 줄이고 페이지 처리 시간을 개선할 수 있다. 이러한 맥락에서 볼 때, GraphQL은 효율성과 성능 측면에서 REST API에 비해 유리한 조건을 갖추고 있다고 할 수 있다.

반면, REST API는 대표적인 웹 서비스 인터페이스 기술로, 정해진 규약에 따라 데이터를 구조화하여 제공한다. 이는 개발자가 예측 가능하고 일관된 방식으로 데이터를 처리할 수 있게 하지만, 클라이언트가 필요로 하는 데이터보다 많거나 적은 정보를 전송하게 되는 과잉 또는 미달 전송의 문제를 갖는다. 이에 따라 페이지 로딩 시간에 영향을 줄 수 있으며, 특히 대용량 데이터를 다루는 경우 불필요한 지연이 발생할 수 있다.

그러나 Table 2에서 제시하는 결과는 단순히 페이지 로딩 시간의 차이만을 나타내는 것이 아니라, 복잡하고 다양한 쿼리를 요구하는 현대의 애플리케이션에 대해 두 기술이 어떻게 대응하는지에 대한 통찰을 제공한다. 따라서 이 데이터는 특정 상황과 요구 사항에 가장 적합한 데이터 인터페이스 기술을 선택하는 데 있어 중요한 기준을 제공한다.

### 3.1 데이터 인터페이스 다이어그램

본 연구는 조선해양기자재 제작을 위한 SCM시스템과 기업 자원 계획 시스템(ERP) 간의 데이터 인터페이스 기술로 Fig. 3은 두 시스템 간의 데이터 통신 프로세스를 그래픽으로 보여준다.

먼저, SCM시스템은 REST API를 통해 데이터를 요청하고 응답받는다. 이 API는 OAuth2 인증 방식을 사용하여 보안성을 높였다. 클라이언트는 특정 API를 통해 데이터 요청(HTTP Request)을 보내며, 이에 대한 응답으로 필요한 데이터(HTTP Response)를 받는다.

이 데이터는 60초의 주기로 갱신되며, 갱신 데이터는 "I/F 데이터랩"을 통해 ERP 시스템으로 전송된다. ERP 시스템은 받은 데이터를 업데이트하고 저장한다. 이러한 프로세스를 통해 두 시스템 간의 실시간 데이터 동기화가 가능하게 되었다.

이 인터페이스 기술의 도입으로 조선해양기자재 제작 프로세스의 효율성이 크게 향상되었으며, 데이터의 실시간 처리로 인한 업무 처리 시간 단축 및 오류 감소 효과가 있었다.

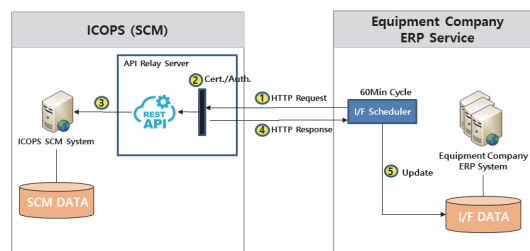


Fig. 3 Data Interface Diagram

### 3.2 실시간 데이터 인터페이스 구조 제안

Fig. 4는 API 기반의 데이터 인터페이스 프로세스를 설명하는 다이어그램이다. 먼저, 클라이언트

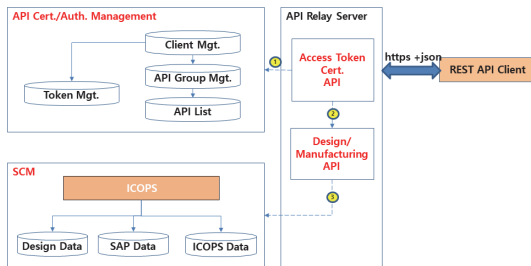


Fig. 4 Real-time data interface structure

는 “Token 요청”을 통해 API 서버에 액세스 토큰을 요청한다. 이 요청이 성공하면, API 서버는 “Access Token 발급” 단계를 통해 클라이언트에게 액세스 토큰을 제공한다. 액세스 토큰을 보유한 클라이언트는 이 토큰을 사용하여 API 서버에 데이터 요청을 할 수 있다.

API 요청이 성공적으로 이루어지면, 데이터는 SCM 시스템 내의 여러 부분에서 가져올 수 있다. 여기에는 “원자재 Data”, “SAP Data”, 그리고 “ICOPS Data”의 세 가지 주요 데이터 소스가 포함된다.

이 프로세스는 데이터 요청과 응답이 실시간으로 이루어지며, 클라이언트는 REST API Client를 통해 해당 데이터를 받아올 수 있다. 전반적으로 이 다이어그램은 API를 통한 실시간 데이터 인터페이스의 중요한 과정을 효과적으로 보여준다.

## 4. 결론

본 연구를 통해 도입될 SCM 기반의 실시간 데이터 인터페이스 기술, 특히 GraphQL과 REST API는 조선해양기자재 제작의 효율성을 극대화할 것으로 기대된다.

GraphQL은 데이터의 오버페칭과 언더페칭을 방지하며, 정확한 데이터 요구 사항을 충족시키는 맞춤형 쿼리를 가능하게 함으로써 네트워크 효율

성을 향상시킬 것이다. 따라서 다양한 종류의 데이터 정보를 하나의 요청으로 가져오도록 하는 웹 애플리케이션과 모바일 애플리케이션에 적용하면 효율적이다.

REST API는 균일하고 예측 가능한 데이터 구조를 통해 빠른 개발과 간단한 관리를 제공한다. 즉, 공정정보, 자재정보, BOM 등과 같이 Format이 일정한 데이터의 인터페이스 구현에 장점이 있다.

두 개의 API를 활용하는 하이브리드형태로 3D 가시화와 BOM을 연동하는 시스템 구현에 활용이 가능하다. 즉, 이 두 기술의 특성을 조합한다면, 조선해양기자재의 공급망 플랫폼에서 데이터 처리 및 관리가 더욱 효과적이고 유연해질 것으로 예상된다.

끝으로 본 연구가 데이터 인터페이스의 표준을 정의하는 과정으로 향후 연구에서는 더욱 다양한 Case Study를 통하여 좀 더 최적화할 수 있는 방안을 도출하고자 한다.

## 후 기

이 논문은 산업통상자원부 ‘산업전문인력역량강화사업’의 재원으로 한국산업기술진흥원(KIAT)의 지원을 받아 수행된 연구임(2023년 미래 해양플랜트 글로벌 고급 전문인력 양성사업, 과제번호: P0012646).

## 참고문헌

- [1] Smith, J. A., & Jones, M. B. (2019). “The role of supply chain management in modern business”, *International Journal of Business Management*, vol.5, no.2, 123-138.
- [2] Lee, K. T., & Choi, Y. S. (2020). “API-based real-time data interface for SCM systems.”,

- Journal of System Integration, vol.7, no.1, 45-59.
- [3] Park, H. J., & Kim, S. H. (2018). "Integration challenges in supply chain systems: A review". Supply Chain Quarterly, vol.14, no.3, 220-234.
- [4] Rodriguez, M. L., & Garcia, P. (2017). "OAuth2 authentication for secured API data access.", Journal of Web Security, vol.9, no.4, 300-312.
- [5] Kumar, R., & Gupta, A. (2019). "Modern data management techniques in supply chain systems.", Logistics and Operations Review, vol.8, no.2, 78-90.
- [6] Choi, B. W., & Han, S. J. (2021). "Advantages and limitations of GraphQL in modern data systems.", Database Journal, vol.10, no.1, 10-25.
- [7] Johnson, L. R., & Wilson, G. H. (2020). "RESTful API design and optimization strategies.", Web Systems Review, vol.12, no.3, 156-169.
- [8] Evans, M., & Taylor, D. (2018). "Digital transformation in supply chain management: A case study.", Digital Business Review, vol. 4, no.2, 50-66.

---

(접수: 2024.01.11. 수정: 2024.01.18. 게재확정: 2024.01.26.)