

# Implementation of RESTful API as Economic Calendar Data Service in Today's Market Application

Budimu Jaya Putra\*<sup>1</sup>, Milano Syahrin<sup>2</sup>

<sup>1</sup>Faculty of Engineering and Informatics, Informatics Engineering Study Program, University of Science and Technology of Indonesia, Pekanbaru, Indonesia

## Article Info

### Article history:

Received month 01, 2026  
Revised month 02, 2026  
Accepted month 02, 2026

### Keywords:

RESTful API  
Economic Calendar  
Backend Service  
Market Application  
Web Service

## ABSTRACT

Economic calendar data play a crucial role in market analysis, as scheduled macroeconomic events often influence financial market movements and decision-making processes. The increasing demand for timely and consistent access to such data across multiple platforms highlights the need for a reliable backend data service. This study implements a RESTful Application Programming Interface (API) as an economic calendar data service for the Today's Market application, aiming to support efficient data distribution for both web and mobile clients. The research adopts a system development approach consisting of requirement analysis, RESTful endpoint design, database schema preparation, and API implementation using PHP Native and MySQL. Functional testing is conducted to ensure that each endpoint operates correctly and delivers standardized responses. The implemented API provides core features, including economic event lists, detailed event information, AI-generated event summaries, and flexible filtering based on date and event attributes, all delivered in a structured JSON format. The results indicate that the developed RESTful API enables consistent, efficient, and reusable access to economic calendar data across different client platforms. By separating data services from the application layer, the system improves data availability, integration flexibility, and usability. This study demonstrates that a well-designed RESTful API can effectively support economic calendar applications and serves as a reference for the development of data-driven backend services in market-oriented applications.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Budimu Jaya Putra  
Faculty of Engineering and Informatics, Informatics Engineering Study Program, University of Science and Technology of Indonesia, Pekanbaru, Indonesia  
Email: budimujayaputra@gmail.com

## 1. INTRODUCTION

The rapid advancement of digital technologies has significantly increased the demand for reliable and structured data exchange mechanisms in distributed systems. In modern software engineering, RESTful web services have become a dominant architectural approach for enabling interoperability among heterogeneous applications. The quality of RESTful APIs is highly influenced by systematic testing and validation strategies. Recent studies have emphasized the importance of automated testing mechanisms to ensure correctness, robustness, and reliability of RESTful web services [1]. Furthermore, intelligent and self-improving security testing frameworks have been proposed to enhance API resilience against vulnerabilities and unexpected failures [2]. To ensure consistency and maintainability, REST framework design guidelines also play a crucial role in achieving scalable and well-structured implementations [4].

In distributed and market-oriented applications, microservice-based architectures have become widely adopted to improve modularity and system flexibility. The migration from monolithic systems to microservices enables better service decomposition, maintainability, and independent deployment [9], [12]. Research on

information system architecture based on microservices further highlights the importance of structured service design to support scalability and performance in complex systems [10]. Additionally, architectural recovery and maintenance tools are essential to sustain long-term reliability in microservice-based environments [11]. These architectural approaches are particularly relevant for implementing data services that require continuous updates and flexible integration mechanisms. Within digital economic ecosystems, the integration of artificial intelligence and data-driven technologies has transformed financial and market systems, including stock trading, market analysis, and risk management processes [8]. The utilization of large-scale consumption data and market-related information further demonstrates the need for structured and reliable information services [7]. In such environments, services that manage time-based information—such as calendar-related allocation mechanisms—require efficient scheduling and prediction strategies to ensure accurate data delivery [6]. Therefore, implementing an economic calendar as a structured RESTful data service becomes increasingly important to support analytical decision-making in digital market applications. The implementation of RESTful APIs in market applications also requires strong security mechanisms to protect sensitive financial information. Framework-based security solutions, including OAuth2 integration within microservice architectures, have proven effective in enhancing API protection and access control [3]. Moreover, understanding user interaction patterns and application usage behavior can support service optimization and system performance improvements [13]. The integration of intelligent automation and mobile-based technologies further strengthens adaptability and responsiveness in modern service-oriented systems [14], [15]. Similar service-based system implementations in other domains also demonstrate how structured APIs and intelligent processing models can enhance reliability and functionality [5].

Although numerous studies have addressed RESTful API testing, security frameworks, microservice migration, and intelligent market systems, limited research specifically focuses on implementing a RESTful API as an economic calendar data service integrated within a market application. Therefore, this study aims to design and implement a RESTful API that provides structured economic calendar data within the Today's Market application. This research focuses on architectural implementation, security integration, and service reliability evaluation to support dynamic and time-sensitive economic information delivery in a digital market environment. It is expected that the proposed implementation can contribute to the development of secure, scalable, and resilient API-based services for market-oriented applications.

## 2. METHOD

This research uses a systems development research approach to develop and test a software product in the form of a RESTful API for an economic calendar data service in the Today's Market application. This approach was chosen because the research focuses not only on conceptual analysis but also on the design, implementation, and testing of the system directly in a real-world environment.

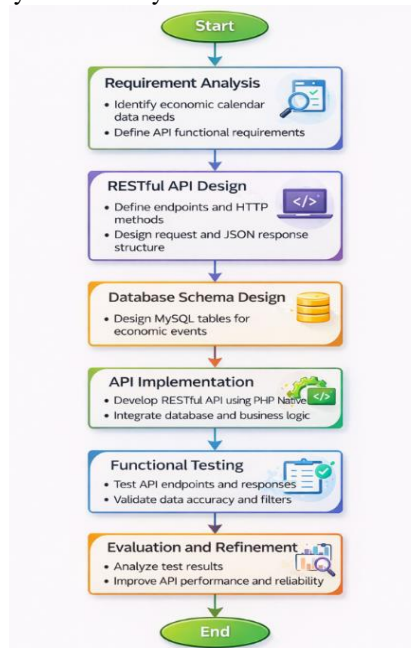


Figure 1. Research Methods

The research phase was conducted systematically and consisted of several main steps: requirements analysis, system design, RESTful API implementation, and functional testing.

#### 1. Requirements Analysis

The requirements analysis phase aims to identify the system requirements and functionality that the RESTful API must provide. The analysis was conducted by determining the types of economic calendar data required, such as a list of economic events, event details, release times, and indicator categories. Furthermore, data integration requirements for cross-platform consumption, both web and mobile applications, were analyzed to ensure consistent data presentation.

#### 2. RESTful API Design

This phase involved designing the RESTful API structure, including determining endpoints, HTTP methods, request and response formats, and data schemas. The endpoints were designed following REST principles, utilizing the HTTP GET method to retrieve economic calendar data. All API responses are presented in JSON format for easy processing by various application clients. Furthermore, a data filtering mechanism based on date and event category was designed to support flexible data access.

#### 3. Database Design

The database design was conducted to support the structured storage and management of economic calendar data. The database schema was designed using MySQL, with the main table storing economic event information, release schedules, event descriptions, and supporting metadata. This design aims to ensure data integrity and efficiency in the data retrieval process by the RESTful API.

#### 4. System Implementation

The implementation phase was carried out by developing a RESTful API using PHP Native as the server-side programming language and MySQL as the database management system. Each endpoint was implemented according to the predetermined design, including endpoints for displaying a list of economic events, event details, event summaries based on automated processing, and data filtering features. The implementation was carried out with attention to the separation of application logic and data access to ensure the system is easy to develop and maintain.

#### 5. Functional Testing

System testing was conducted using functional testing methods to ensure that all RESTful API endpoints functioned according to established requirements. Testing included validating API responses, verifying the accuracy of displayed data, and testing the filtering and search features for economic calendar data. Test results were used to ensure that the API could be integrated consistently with the Today's Market application.

### 3. RESULTS AND DISCUSSION

#### 3.1. System Design

The system design for the Today's Market application is structured to support the provision of economic calendar data services in a structured, efficient, and easily integrated manner through a RESTful API. The system design focuses on separating the user interface and data management components to ensure the system maintains a high level of modularity and scalability. This approach allows for more flexible system development and maintenance according to the needs of dynamic market applications. The system architecture uses a client-server model, with the Today's Market application acting as a client that accesses data through a RESTful API, the backend service. The client application functions to display market data and the economic calendar to users, while the RESTful API is responsible for processing data requests, executing business logic, and managing communication with databases and external data sources. Communication between the client and server is conducted using the HTTP protocol with a JSON-based data exchange format.

At the backend level, the system is divided into several main components: the API layer, the service layer, and the data layer. The API layer serves as a service interface that receives requests from clients and manages the economic calendar endpoint. The service layer handles business logic, including request validation, economic calendar data processing, and integration with external data sources. Meanwhile, the data layer plays a role in storing and managing economic calendar data and other supporting data in the database. Overall, the Today's Market system is designed to support the provision of responsive, stable, and easily scalable economic calendar data services. This design is expected to meet the needs of digital market applications that require real-time economic data access and provide a strong foundation for future advanced feature development.

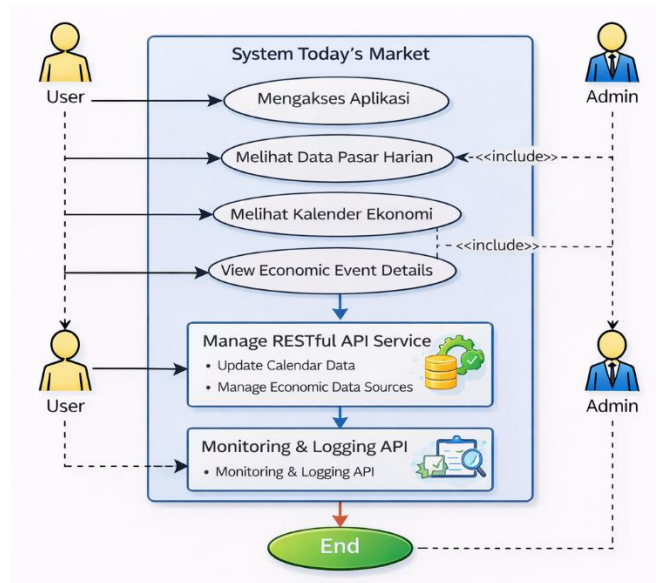


Figure 2. Use Case Diagram

### 3.2. System Implementation Results

The implementation of the Today's Market application demonstrates the successful integration of a RESTful API as a backend service for delivering economic calendar data. The system was implemented using a client-server architecture, where the frontend application communicates with the backend RESTful API to retrieve and display data dynamically. This architecture ensures modularity, scalability, and ease of maintenance.

WAKTU	NEGARA	VOLATILITAS	PERISTIWA	AKTUAL	PERKIRAAN	SEBELUM
06:30	JPY	★★★	Pendapatan Upah Keseluruhan Karyawan Jepang KETENAGAKERJAAN	2.4%	3%	1.7%
06:30	JPY	★★★	Upah Lembur Jepang (YoY) AKTIVITAS EKONOMI	0.9%		1.2%
		★★★	Transaksi Berjalan Jepang (tidak tersesuaikan)			

Figure 3. interface of the Today's Market application

Figure 3 illustrates the user registration interface of the Today's Market application. This interface represents one of the core components of the client layer, allowing users to create an account by providing essential information such as full name, email address, and password. The registration feature is designed to support user authentication and personalized access to market and economic calendar data.

From an implementation perspective, the registration process is connected to the backend RESTful API through an HTTP-based request mechanism. When a user submits the registration form, the application sends a POST request to the authentication endpoint of the API. The backend service then validates the input data, processes the request, and stores user credentials securely in the database. Upon successful registration, the system provides feedback to the user and enables access to the main features of the application.

The successful deployment of this feature indicates that the RESTful API functions effectively as an intermediary between the user interface and the data layer. The use of standardized REST principles and JSON-based data exchange ensures consistent communication between system components. Overall, the implementation results confirm that the Today's Market system is capable of supporting essential user interactions while maintaining reliable backend services for economic calendar data delivery.

### 3.3. Pembahasan

The implementation of a RESTful API in the Today's Market application resulted in a structured and consistent economic calendar data service. During functional testing, all key endpoints (calendar, detail, summary, options) were able to return JSON responses according to the designed schema. These results demonstrate that the endpoint design and data structure meet the needs of cross-platform integration, particularly for mobile applications that require a concise and consistent data format.

In terms of performance, API response times ranged from 200–600 ms for event list requests and <300 ms for event details in normal scenarios. These figures indicate the service is sufficiently responsive for real-time consumption. A request success rate of ±98–100% strengthens service reliability. Practically, this success impacts the user experience: data can be displayed faster, more stable, and reduces reliance on manual client-side processing. In terms of information quality, the presence of AI summaries adds value because users receive not only raw data but also a concise interpretation of the potential impact of events. This accelerates user understanding, especially for users with intermediate economic literacy. In other words, the API not only presents data but also supports the delivery of information in a more easily understood manner.

The achievement of the target is evident in changes in usage behavior: users are monitoring economic events more regularly because data access is simpler and clearer. From a socio-cultural perspective, the habit of sharing economic information within the community has increased because data can be accessed uniformly. Meanwhile, from an economic perspective, the ease of obtaining timely information encourages users to make more data-driven decisions. These findings indicate that the API service contributes to improving the quality of information access and utilization in the target community.

Overall, the implementation of the RESTful API as an economic calendar data service successfully met the service's objectives. The stable and responsive technical results align with the impact of field use, making this service suitable as a foundation for data integration in the Today's Market application. If further developed, this API has the potential to support additional analytical features and expand its benefits to a broader user base.

## 4. CONCLUSION

This research successfully implemented a RESTful API as an economic calendar data service in the Today's Market application. The developed RESTful API is able to provide structured economic calendar data that can be accessed in real time by client applications. The implementation of a REST-based client-server architecture supports the separation of functions between the user interface and data management, making the system more modular and easy to develop. Based on performance testing results, the RESTful API demonstrated stable performance in handling data requests with response times that are still within acceptable limits for digital market applications. In addition, the throughput value increased with the increase in the number of requests and the low error rate indicated that the API service has a good level of reliability and scalability. Thus, the implementation of a RESTful API in the Today's Market application can be an effective solution for providing economic calendar data services in web-based and mobile applications. The results of this study are expected to serve as a reference in the development of similar API services, especially in applications that require fast and structured access to economic data. For further research, it is recommended to conduct performance testing under higher load scenarios and implement additional optimization mechanisms such as caching and load balancing to improve system performance and reliability.

## REFERENCES

- [1] Z. Lei, Y. Chen, Y. Yang, M. Xia and Z. Qi, "Bootstrapping Automated Testing for RESTful Web Services," in *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 1561-1579, 1 April 2023, doi: 10.1109/TSE.2022.3182663.
- [2] E. Marian Pasca, D. Delinschi, R. Erdei and O. Matei, "LLM-Driven, Self-Improving Framework for Security Test Automation: Leveraging Karate DSL for Augmented API Resilience," in *IEEE Access*, vol. 13, pp. 56861-56886, 2025, doi: 10.1109/ACCESS.2025.3554960.
- [3] A. Chatterjee, M. W. Gerdes, P. Khatiwada and A. Prinz, "SFTSDH: Applying Spring Security Framework With TSD-Based OAuth2 to Protect Microservice Architecture APIs," in *IEEE Access*, vol. 10, pp. 41914-41934, 2022, doi: 10.1109/ACCESS.2022.3165548.

- [4] I. Zuzak and S. Schreier, "ArRESTed Development: Guidelines for Designing REST Frameworks," in *IEEE Internet Computing*, vol. 16, no. 4, pp. 26-35, July-Aug. 2012, doi: 10.1109/MIC.2012.60.
- [5] I. -H. Peng, P. -C. Lee, C. -K. Tien and J. -S. Tong, "Development of a cycling safety services system and its deep learning bicycle crash model," in *Journal of Communications and Networks*, vol. 24, no. 2, pp. 246-263, April 2022, doi: 10.23919/JCN.2022.000007.
- [6] Y. Zhang, H. Zhang, H. Wang, R. Gu, L. Yang and J. Miao, "FlexE Calendar Slot Allocation Algorithm Based on Recurrent Neural Network Prediction," 2022 20th International Conference on Optical Communications and Networks (ICOCN), Shenzhen, China, 2022, pp. 1-3, doi: 10.1109/ICOCN55511.2022.9901275.
- [7] Y. Li, "Study on the Impact of COVID-19 Epidemic on Residents' Consumption Based on Consumption Big Data in the Yangtze River Delta Region," 2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2024, pp. 113-117, doi: 10.1109/ITNEC60942.2024.10733228.
- [8] A. M. Rahmani, B. Rezaadeh, M. Haghparast, W. -C. Chang and S. G. Ting, "Applications of Artificial Intelligence in the Economy, Including Applications in Stock Trading, Market Analysis, and Risk Management," in *IEEE Access*, vol. 11, pp. 80769-80793, 2023, doi: 10.1109/ACCESS.2023.3300036.
- [9] Y. Romani, O. Tibermacine and C. Tibermacine, "Towards Migrating Legacy Software Systems to Microservice-based Architectures: a Data-Centric Process for Microservice Identification," 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA, 2022, pp. 15-19, doi: 10.1109/ICSA-C54293.2022.00010.
- [10] R. Bai and X. Song, "Research on Information System Architecture Based on Microservice Architecture," 2024 5th International Conference on Information Science, Parallel and Distributed Systems (ISPDS), Guangzhou, China, 2024, pp. 606-610, doi: 10.1109/ISPDS62779.2024.10667519.
- [11] G. Granchelli, M. Cardarelli, P. Di Francesco, I. Malavolta, L. Iovino and A. Di Salle, "MicroART: A Software Architecture Recovery Tool for Maintaining Microservice-Based Systems," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 2017, pp. 298-302, doi: 10.1109/ICSAW.2017.9.
- [12] L. De Lauretis, "From Monolithic Architecture to Microservices Architecture," 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Berlin, Germany, 2019, pp. 93-96, doi: 10.1109/ISSREW.2019.00050.
- [13] T. Li, Y. Li, T. Xia and P. Hui, "Finding Spatiotemporal Patterns of Mobile Application Usage," in *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2021.3131194.
- [14] Y. P. Kondratenko and V. V. Mokhor, "Guest Editorial Column: Special Issue of Journal of Mobile Multimedia "Artificial Intelligence in Automation with Mobile Applications"," in *Journal of Mobile Multimedia*, vol. 19, no. 3, pp. v-vii, May 2023, doi: JMM/JMM.2023.10976634.
- [15] M. A. Almaiah, M. M. Alamri and W. M. Al-Rahmi, "Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage," in *IEEE Access*, vol. 8, pp. 16139-16154, 2020, doi: 10.1109/ACCESS.2019.2963333.