

# Brooke

## API Portal Implementation Considerations



### Description

Brooke supports clients to design, configure and rapidly implement Application Programming Interface (API) portals to improve the way APIs of all types are catalogued, consumed and managed.

Brooke believes in solving business problems with technology as the enabler, rather than starting with the technology task first.

By building a common API catalog using MuleSoft's Anypoint API Community Manager, Brooke establishes a mechanism to discover, consume and manage APIs across all development efforts. The portal supports the client to catalog all APIs, agnostic of platform, relevant to the efficient execution of business processes. Initial registration can be self-service or access controlled, depending on the client's governance requirements. Trusted users can then securely self-serve API details as required.

The Brooke team approaches all client engagement with a consultative and collaborative approach, focused on building our clients' capability and supporting them to achieve their broader objectives.

### Implementation and Operation Considerations

Critical considerations when designing, configuring and implementing an API portal help to ensure the portal's effectiveness, usability and security.

Brooke regards the following as the essential aspects for ICT teams to consider:

1. **API Discovery and Documentation:** Ensure the API portal provides comprehensive documentation for each API, which is version controlled. Include detailed descriptions, usage guidelines, request/response examples, and any necessary authentication or authorisation requirements. Implement robust search capabilities and categorisation to enable easy discovery of APIs.



2. **User Experience:** Customise the user experience to enhance stakeholder engagement and ease of API consumption.
3. **Developer Registration and Onboarding:** Implement a seamless developer registration process that allows developers to create accounts, obtain API keys or access tokens, and track their API usage. Provide clear instructions and resources to guide developers through the onboarding process.
4. **API Testing:** Include a mocking service in the API Portal to allow developers to test APIs.
5. **API Versioning and Lifecycle Management:** Support API versioning to manage changes and updates to APIs. Communicate versioning policies and provide mechanisms for developers to access and migrate to newer versions. Consider incorporating API lifecycle management features, such as deprecation and retirement processes.
6. **Security and Access Control:** Implement strong security measures to protect APIs and sensitive data. Include authentication and authorisation mechanisms (e.g., API keys, OAuth) to ensure only authorised developers and applications can access the APIs. Consider implementing rate limiting and throttling to prevent abuse and ensure fair usage.
7. **Developer Support and Community Engagement:** Offer developers support channels, such as FAQs, forums, and ticketing systems, to address developer inquiries and provide assistance. Foster a sense of community by encouraging developer collaboration, sharing best practices, and organising developer events or hackathons.
8. **Mobile-Optimized and Responsive Design:** Ensure that the API portal is designed to be mobile-friendly and accessible across different devices and screen sizes; this allows developers to seamlessly utilise the portal from desktops, laptops and tablets, and ensures API information is consumable via mobile devices.
9. **Integration with Developer Tools and IDEs:** Consider integrating the API portal with popular developer tools, such as Integrated Development Environments (IDEs), code repositories, and Continuous Integration/Continuous Deployment (CI/CD) pipelines; this integration enhances developer productivity and streamlines the development and deployment process.
10. **Compliance and Data Privacy:** Ensure compliance with relevant data protection and privacy regulations, such as General Data Protection Regulation (GDPR) and other government standards, such as National API Design Standards (NAPIDS) and state guidance such as the Victorian Protective Data Security Standards (VPDSS). Implement appropriate data handling practices and provide data usage and privacy policy transparency.
11. **Operating Model:** Ensure operating model considerations have been applied to support and maintain the different areas of the portal, such as training, support, knowledge articles and how-to guides, deployment and governance. Establish the long-term viability of the portal by embedding it into BAU operations.

---

## Rapid Transformation | Meaningful Results

Learn more about the benefits of partnering with Brooke to solve your complex business problems by contacting Bruce McGregor, Managing Partner, at [brucemcgregor@brooke.global](mailto:brucemcgregor@brooke.global) to discuss.

