The Addison-Wesley Signature Series

A Vaughn Vernon Signature Book

Principles of Web API Design

Delivering Value with APIs and Microservices

James Higginbotham

Foreword by
Mike Amundsen

Praise for *Principles of Web API Design*

"I've had the good fortune to work alongside and learn from James over the past several years. His varied institutional knowledge, along with his depth of experience and eye for practical application, makes him unique among his peers. I am ecstatic that others now have the opportunity, in this book, to benefit from James's compelling, pragmatic vision for how to make better APIs. *Principles of Web API Design* surveys the gamut of available techniques and sets forth a prescriptive, easy-to-follow approach. Teams that apply the guidance in this book will create APIs that better resonate with customers, deliver more business value in less time, and require fewer breaking changes. I cannot recommend *Principles of Web API Design* enough."

—Matthew Reinbold, Director of API Ecosystems, Postman

"James is one of the preeminent experts on API design in the industry, and this comprehensive guide reflects that. Putting API design in the context of business outcomes and digital capabilities makes this a vital guide for any organization undergoing digital transformation."

—Matt McLarty, Global Leader of API Strategy at MuleSoft, a Salesforce company

"In modern software development, APIs end up being both the cause of and solution to many of the problems we face. James's process for dissecting, analyzing, and designing APIs from concepts to caching creates a repeatable approach for teams to solve more problems than they create."

—D. Keith Casey, Jr., API Problem Solver, CaseySoftware, LLC

"Following James's clear and easy-to-follow guide, in one afternoon I was able to apply his process to current real-world use cases. I now have the practical guidance, techniques, and clear examples to help me take those next vital steps. Recommended reading for anyone connected to and working with APIs."

—Joyce Stack, Architect, Elsevier

"*Principles of Web API Design* uncovers more than principles. In it, you'll learn a process—a method to design APIs."

—Arnaud Lauret, API Handyman

"This insightful playbook guides API teams through a structured process that fosters productive collaboration, valuable capability identification, and best-practice contract crafting. James distills years of experience into a pragmatic roadmap for defining and refining API products, and also provides a primer for API security, eventing, resiliency, and microservices alignment. A must-read for architects either new to the API discipline or responsible for onboarding new teams and instituting a structured API definition process."

—Chris Haddad, Chief Architect, Karux LLC

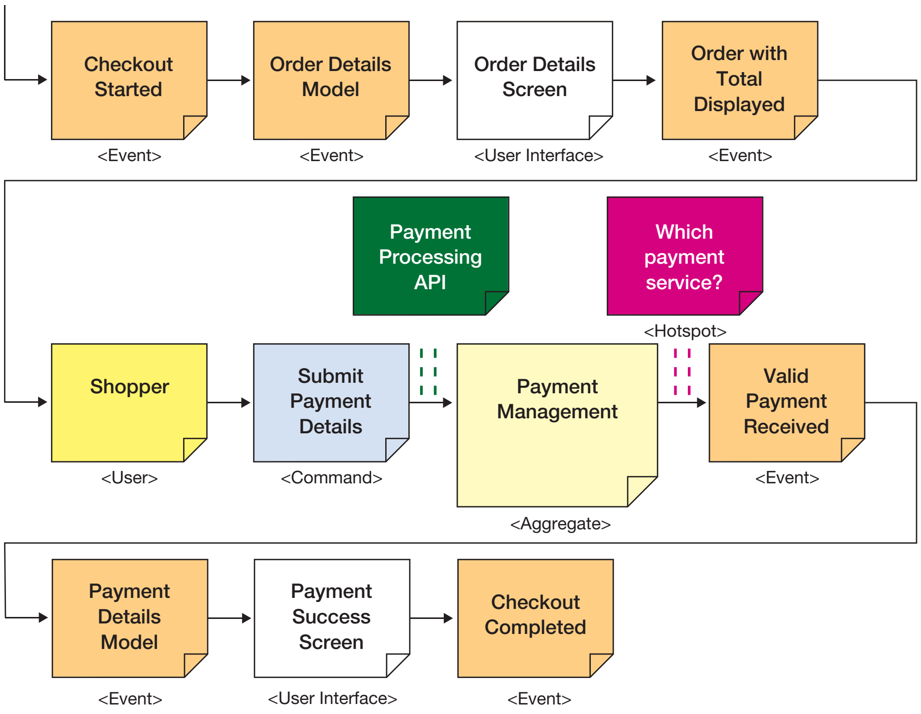

Figure 5.3 *(continued)*

Table 5.1 *Activity Steps for JSON's Bookstore; Separators Indicate Shifts in Vocabulary That Identify Boundaries*

Digital Capability	Activity	Activity Step	Participants	Description
Place an Order	Browse for Books	List Books	Customer, Call Center	List books by category or release date
Place an Order	Browse for Books	Search for Books	Customer, Call Center	Search for books by author, title
Place an Order	Browse for Books	View Book Details	Customer, Call Center	View the details of a book
Place an Order	Shop for Books	Add Books to Cart	Customer, Call Center	Add a book to the customer's cart
Place an Order	Shop for Books	Remove Books from Cart	Customer, Call Center	Remove a book from the customer's cart
Place an Order	Shop for Books	Clear Cart	Customer, Call Center	Remove all books from the customer's cart
Place an Order	Shop for Books	View Cart	Customer, Call Center	View the current cart and total
Place an Order	Create an Order	Checkout	Customer, Call Center	Create an order from the contents of the cart
Place an Order	Create an Order	Pay for Order	Customer, Call Center	Accept and process payment for the order

Avoid using the terms *service* and *manager*, as they are generally not useful in understanding the purpose of the API.

The APIs in Figure 5.3 are named Shopping API, Order Creation API, and Payment Processing API. This is a good start and clearly articulates the scope and responsibility of each API.

Note

Some API designers may prefer to combine the Order Creation and Payment Processing APIs, as they could be considered cohesive and therefore should exist as a single API. They are separated in this simple example for instructional purposes. However, field insights dictate that separating order creation and payment through clearly defined boundaries allows for more complex payment processing at a future time without burdening the order creation boundary with the added complexity.

Finally, separate the activity steps into the corresponding API based on the boundary it represents. Table 5.2 captures the Shopping API, including the activity steps relevant to the API.

Table 5.3 captures the checkout process for the Place an Order digital capability.

Finally, Table 5.4 captures the payment step as part of the Place an Order digital capability.

Table 5.2 *Shopping API, Discovered through Boundary Identification, with Corresponding Activity Steps from JSON's Bookstore*

Digital Capability	Activity	Activity Step	Participants	Description
Place an Order	Browse for Books	List Books	Customer, Call Center	List books by category or release date
Place an Order	Browse for Books	Search for Books	Customer, Call Center	Search for books by author, title
Place an Order	Browse for Books	View Book Details	Customer, Call Center	View the details of a book
Place an Order	Shop for Books	Add Books to Cart	Customer, Call Center	Add a book to the customer's cart
Place an Order	Shop for Books	Remove Books from Cart	Customer, Call Center	Remove a book from the customer's cart
Place an Order	Shop for Books	Clear Cart	Customer, Call Center	Remove all books from the customer's cart
Place an Order	Shop for Books	View Cart	Customer, Call Center	View the current cart and total
Place an Order	Create an Order	Checkout	Customer, Call Center	Create an order from the contents of the cart
Place an Order	Create an Order	Pay for Order	Customer, Call Center	Accept and process payment for the order

Table 5.3 *Order Creation API, Discovered through Boundary Identification, with Corresponding Activity Steps from JSON's Bookstore*

Digital Capability	Activity	Activity Step	Participants	Description
Place an Order	Create an Order	Checkout	Customer, Call Center	Create an order from the contents of the cart

Table 5.4 *Payment Processing API, Discovered through Boundary Identification, with Corresponding Activity Steps from JSON's Bookstore*

Digital Capability	Activity	Activity Step	Participants	Description
Place an Order	Create an Order	Pay for Order	Customer, Call Center	Accept and process payment for the order

With the boundaries clearly defined, API modeling can begin. This effort leads to API profiles that define the operations and events each API will offer. API modeling is detailed in the next chapter.

Summary

APIs benefit from careful scoping and assignment of responsibilities. Applying boundaries helps to identify one or more APIs that will be required to deliver the desired outcomes captured as job stories. This prepares the way for the next step, API modeling, in which a blueprint of the API is formed and the foundation is laid for API design.

Chapter 6

API Modeling

You can use an eraser on the drafting table or a sledgehammer on the construction site.

—Frank Lloyd Wright

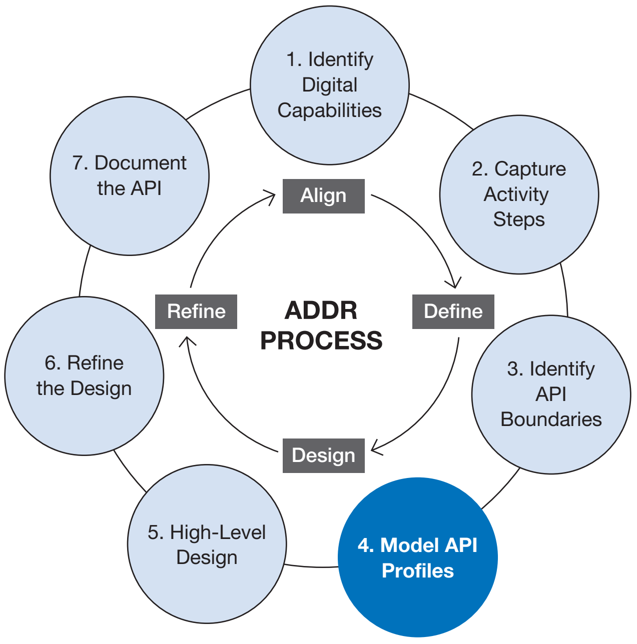

Figure 6.1 *The final step of the Define phase is to create API profiles in preparation for transitioning to the Design phase.*

Developers are often tempted to start writing code immediately. Code is the primary tool for developers. It is everything—the hammer, screwdriver, ruler, and saw. When code is seen as the one and only tool to design an API, the quality of the API design can suffer. The march to produce code for production becomes more valued than the outcomes the API is meant to produce.

Of course, code produces value when it is used to explore a specific area of a solution to reduce risk. It is also valuable to use code to experiment, surface unknowns, or explore a new technology. The term *tracer bullet* was applied to software by David Thomas and Andrew Hunt in their book *The Pragmatic Programmer*.[1] The term describes the use of code as a means of exploration and risk reduction. Tracer bullet code delivers value through learning rather than through production-ready code.

API modeling is a tracer bullet for API design. It is a technique for exploring the necessary elements of an API prior to the design and delivery process. API modeling (see Figure 6.1) helps to bring together the insights and artifacts from the previous steps into an API profile that describes the scope and intent of the APIs needed to deliver the desired outcomes of end users.

What Is API Modeling?

Just as a beautiful Web design begins from a wireframe, a great API design begins with an API model that helps define its scope and responsibilities. The goal of API modeling is to fully understand and validate the needs of developers and end users. Unlike a wireframe, which focuses strictly on end user interaction, API modeling focuses on both developer and end user goals. Often, these goals are aligned, but sometimes they are not. API modeling helps to surface issues quickly so that they may be resolved prior to writing code.

API modeling uses job stories, activities, and activity steps as inputs to produce a cohesive view of each API, called an *API profile*. An API profile captures characteristics about the API, including its name, scope, operations, and emitted events that will be used to deliver desired outcomes. API modeling is done before designing and developing begins—while the cost of change is significantly lower.

After completing API modeling, teams will be ready to migrate the API profiles produced into an API design. API modeling can be used as input for a single API design style of choice, such as REST, GraphQL, or gRPC. It may also be used to

1. David Thomas and Andrew Hunt, *The Pragmatic Programmer: Your Journey to Mastery*, 20th Anniversary Edition, 2nd ed. (Boston: Addison-Wesley, 2020).

inform the design of an API that uses a combination of these API styles to support the various digital channels for customers and partner integration needs.

The API Profile Structure

API profiles capture all necessary information about an API, independent of the API style or styles that it will expose (e.g., REST and GraphQL). The API profile is used to drive the design of an API, but also provides the beginnings of the API documentation effort during the early stages of API definition.

An API profile captures the following details about each API:

- The name and a short description of the API
- The scope of the API (internal, public, partner, etc.)
- API operations with input and output message details
- Participants allowed to perform each operation, in preparation for securing the API
- Events generated by each API operation, to drive extensibility beyond the API's original intent
- (Optional) architectural requirements identified, such as a service-level agreement (SLA)

A spreadsheet or document is sufficient to capture each API profile. Using a collaborative spreadsheet allows teams to capture and refine API profiles without the need to email changes among team members. Some teams prefer to use tools such as wikis for capturing API profiles. No matter what tool is selected, be sure that everyone in the organization has access to read and comment on the API profiles produced. Using a tool that is provisioned for only a subset of the organization is not recommended.

Figure 6.2 shows a template that is easy to read and fits both spreadsheet and document formats.

The API Modeling Process

The goal of the API modeling process is to produce one or more API profiles, one for each API identified during modeling. The modeling process is divided into five quick steps. Each step adds additional detail to the API profile until a blueprint of the API emerges.