

Curriculum for

Certified Professional for
Software Architecture (CPSA)[®]
Advanced Level

Module
API

Application Programming Interfaces

2025.1-rev1-EN-20250306



Table of Contents

List of Learning Goals	2
Introduction: General information about the iSAQB Advanced Level	3
What is taught in an Advanced Level module?	3
What can Advanced Level (CPSA-A) graduates do?	3
Requirements for CPSA-A certification	3
Essentials	4
What does the module “API” convey?	4
Curriculum Structure and Recommended Durations	4
Duration, Teaching Method and Further Details	4
Prerequisites	5
Structure of the Curriculum	5
Supplementary Information, Terms, Translations	5
1. Why APIs are Important	6
1.1. Terms and Principles	6
1.2. Learning Goals	6
1.3. References	6
2. How APIs are Creating Value	7
2.1. Terms and Principles	7
2.2. Learning Goals	7
2.3. References	7
3. API Styles and Technologies	8
3.1. Terms and Principles	8
3.2. Learning Goals	8
3.3. References	8
4. API Design	9
4.1. Terms and Principles	9
4.2. Learning Goals	9
4.3. References	10
5. Description of APIs	11
5.1. Terms and Principles	11
5.2. Learning Goals	11
5.3. References	11
6. API Lifecycle and API Tooling	12
6.1. Terms and Principles	12
6.2. Learning Goals	12
6.3. References	12
7. API Security	13



7.1. Terms and Principles 13

7.2. Learning Goals 13

7.3. References..... 13

8. APIs at Scale: Platforms and Governance 14

8.1. Terms and Principles 14

8.2. Learning Goals 14

8.3. References..... 14

References 15

© (Copyright), International Software Architecture Qualification Board e. V. (iSAQB® e. V.) 2025

The curriculum may only be used subject to the following conditions:

1. You wish to obtain the CPSA Certified Professional for Software Architecture Foundation Level® certificate or the CPSA Certified Professional for Software Architecture Advanced Level® certificate. For the purpose of obtaining the certificate, it shall be permitted to use these text documents and/or curricula by creating working copies for your own computer. If any other use of documents and/or curricula is intended, for instance for their dissemination to third parties, for advertising etc., please write to info@isaqb.org to enquire whether this is permitted. A separate license agreement would then have to be entered into.
2. If you are a trainer or training provider, it shall be possible for you to use the documents and/or curricula once you have obtained a usage license. Please address any enquiries to info@isaqb.org. License agreements with comprehensive provisions for all aspects exist.
3. If you fall neither into category 1 nor category 2, but would like to use these documents and/or curricula nonetheless, please also contact the iSAQB e. V. by writing to info@isaqb.org. You will then be informed about the possibility of acquiring relevant licenses through existing license agreements, allowing you to obtain your desired usage authorizations.

Important Notice

We stress that, as a matter of principle, this curriculum is protected by copyright. The International Software Architecture Qualification Board e. V. (iSAQB® e. V.) has exclusive entitlement to these copyrights.

The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal entity according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

List of Learning Goals

- LG 1-1: Understand APIs in the context of software development
- LG 1-2: Compare different styles and concepts of integration
- LG 2-1: Understand the concept of "API Economy"
- LG 2-2: Know various ways of API value creation
- LG 2-3: Understand API business models
- LG 2-4: Understand APIs and digital transformation
- LG 3-1: Understand fundamental API styles
- LG 3-2: Understand popular API technologies and associate them with API styles
- LG 3-3: Understand when to use which style and technology and the consequences of that choice
- LG 4-1: Understand the significance of API design
- LG 4-2: Use the "outside-in" approach for API design
- LG 4-3: Understand openness and extensibility as important aspects for API evolution
- LG 4-4: Understand API versioning
- LG 4-5: Know good practices for API design
- LG 5-1: Understand the significance of API descriptions
- LG 5-2: Understand OpenAPI as a description language for HTTP APIs
- LG 5-3: Understand OpenAPI's structure
- LG 5-4: Understand alternative description languages
- LG 6-1: Understand the API lifecycle
- LG 6-2: Manage APIs as products
- LG 6-3: Know API lifecycle tooling
- LG 7-1: Understand communications security fundamentals
- LG 7-2: Understand security technologies for APIs
- LG 7-3: Understand OWASP API Security Top 10
- LG 8-1: Compare various notions of platform
- LG 8-2: Understand API guidelines
- LG 8-3: Establish and manage API guidelines
- LG 8-4: Use APIs as team interfaces

Introduction: General information about the iSAQB Advanced Level

What is taught in an Advanced Level module?

The module can be attended independently of a CPSA-F certification.

- The iSAQB Advanced Level offers modular training in three areas of competence with flexibly designable training paths. It takes individual inclinations and priorities into account.
- The certification is done as an assignment. The assessment and oral exam is conducted by experts appointed by the iSAQB.

What can Advanced Level (CPSA-A) graduates do?

CPSA-A graduates can:

- Independently and methodically design medium to large IT systems
- In IT systems of medium to high criticality, assume technical and content-related responsibility
- Conceptualize, design, and document actions to achieve quality requirements and support development teams in the implementation of these actions
- Control and execute architecture-relevant communication in medium to large development teams

Requirements for CPSA-A certification

- Successful training and certification as a Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- At least three years of full-time professional experience in the IT sector; collaboration on the design and development of at least two different IT systems
 - Exceptions are allowed on application (e.g., collaboration on open source projects)
- Training and further education within the scope of iSAQB Advanced Level training courses with a minimum of 70 credit points from at least three different areas of competence
- Successful completion of the CPSA-A certification exam



Essentials

What does the module “API” convey?

Application Programming Interfaces, or APIs, enable the use of digital services via programmable interfaces. Their use has increased significantly over the last 20 years. From a software architecture perspective, APIs are relevant from both a producer and consumer point of view: Software components should ideally be understood as reusable products; APIs help to make a software component easily reusable. Today, more and more services are available via APIs. As a consumer of these services, it is easier to concentrate on core tasks and outsource other tasks to external components.

The module looks at the different ways in which APIs create value: as technical interfaces that allow different components to communicate with each other; as organisational interfaces that allow different teams to develop their respective components with fewer interdependencies; and as business-oriented building blocks that allow the organisation to develop new services faster and more effectively.

In this module, we always refer to APIs as network-based interfaces, i.e. not as local programming interfaces.

In addition to the focus on technical topics, strategic aspects such as value creation and the use of APIs as team collaboration in the company are also covered. The module is therefore well suited to approaching the topic of APIs from a broader perspective than a purely technical one.

Curriculum Structure and Recommended Durations

Content	Recommended minimum duration (minutes)	Übungszeit (Minuten)
1. Why APIs are Important	60	0
2. How APIs are Creating Value	60	30
3. API Styles and Technologies	60	30
4. API Design	90	30
5. Description of APIs	90	30
6. API Lifecycle and API Tooling	60	30
7. API Security	60	30
8. APIs at Scale: Platforms and Governance	60	0
Total	540 (9 h)	180 (3 h)

Duration, Teaching Method and Further Details

The times stated below are recommendations. The duration of a training course on the API module should be at least 2 days, but may be longer. Providers may differ in terms of duration, teaching method, type and structure of the exercises, and the detailed course structure. In particular, the curriculum provides no specifications on the nature of the examples and exercises.

Licensed training courses for the API module contribute the following credit points towards admission to the final Advanced Level certification exam:

Methodical Competence:	10 Points
Technical Competence:	10 Points
Communicative Competence:	0 Points

Prerequisites

Participants should have the following knowledge and/or experience:

- Work as an architect or in comparable conceptual or management roles.
- Work as a developer or in comparable technical roles.
- Technically oriented professionals with basic knowledge of IT and software architecture who want to gain a comprehensive overview of APIs.

Structure of the Curriculum

The individual sections of the curriculum are described according to the following structure:

- **Terms/principles:** Essential core terms of this topic.
- **Teaching/practice time:** Defines the minimum amount of teaching and practice time that must be spent on this topic or its practice in an accredited training course.
- **Learning goals:** Describes the content to be conveyed including its core terms and principles.

This section therefore also outlines the skills to be acquired in corresponding training courses.

Supplementary Information, Terms, Translations

To the extent necessary for understanding the curriculum, we have added definitions of technical terms to the [iSAQB glossary](#) and complemented them by references to (translated) literature.

1. Why APIs are Important

Duration: 60 min	Practice time: 0 min
------------------	----------------------

1.1. Terms and Principles

local APIs, network-based APIs, API landscape, systems integration

1.2. Learning Goals

LG 1-1: Understand APIs in the context of software development

Participants understand APIs in the context of programming, computer networks, distributed systems, and software architecture. They understand the development from local APIs to network-based APIs and the current API landscape in the overall context.

LG 1-2: Compare different styles and concepts of integration

Different approaches for systems integration are known to participants and can be compared and contrasted, specifically:

- Integration through a shared database
- File-based systems integration
- Integration through synchronous communications, for example RPC (Remote Procedure Call)
- Integration through asynchronous communications, for example message queues

1.3. References

[\[Fishman and McLarty 2024\]](#), [\[Hohpe and Woolf 2003\]](#)

2. How APIs are Creating Value

Duration: 60 min	Practice time: 30 min
------------------	-----------------------

2.1. Terms and Principles

API economy, API value creation, API business models, digital transformation

2.2. Learning Goals

LG 2-1: Understand the concept of "API Economy"

Participants understand the broader concept of "API Economy" and understand how their products and activities fit into it.

LG 2-2: Know various ways of API value creation

Participants know and understand the various ways in which APIs can help with value creation. On the top level, the different ways can be categorized into three areas:

- Private APIs which are used inside an organization
- Partner APIs which are used with established partners
- Public APIs which are offered as products to the public

Participants understand the various options inside these categories and the relationships between them. They also understand how a comprehensive API strategy adds value for all of these categories.

LG 2-3: Understand API business models

Participants have a detailed knowledge of the various business models for APIs and know a few specific examples. Starting from these different business models, participants understand how to analyze existing business models and how they can be augmented with APIs.

LG 2-4: Understand APIs and digital transformation

Participants know how to fit APIs into the bigger picture of digital transformation. APIs are identified as necessary (but not sufficient) aspect of digital transformation. Other necessary aspects are known as well:

- Business level: creating new channels, exploring new opportunities, being more open to experimentation and customization
- Organizational level: accelerating processes, reducing dependencies between teams and providing a platform that helps teams work more effectively

2.3. References

[\[API Business Models\]](#), [\[Goyal 2023\]](#)

3. API Styles and Technologies

Duration: 60 min	Practice time: 30 min
------------------	-----------------------

3.1. Terms and Principles

RPC, resource, hypermedia, query, events, asynchronous communications, gRPC, HTTP APIs, REST, GraphQL

3.2. Learning Goals

LG 3-1: Understand fundamental API styles

The following API styles are understood by participants and they can balance their advantages against their drawbacks:

- RPC (Remote Procedure Call)
- Resource-based
- Hypermedia
- Query
- Events & asynchronous communications

LG 3-2: Understand popular API technologies and associate them with API styles

Participants are able to classify the most important API technologies and know which API style they are implementing, for example:

- gRPC
- HTTP APIs
- REST
- GraphQL

LG 3-3: Understand when to use which style and technology and the consequences of that choice

Participants gain knowledge of criteria when certain styles and technologies are a good or a bad fit, and which consequences the choice will have.

3.3. References

[\[Hohpe and Woolf 2003\]](#), [\[Porcello 2018\]](#), [\[Richardson et al. 2013\]](#)

4. API Design

Duration: 90 min	Practice time: 30 min
------------------	-----------------------

4.1. Terms and Principles

API design, API first, consumer-driven API design, Postel's law, forward and backward compatibility, versioning and lifecycle of API products, JSON:API, HAL, API design style guides, HTTP, HTTP caching

4.2. Learning Goals

LG 4-1: Understand the significance of API design

Participants understand why careful API design is important and which influence it has on usability, maintenance, evolution, and adoption. Participants understand that providers and consumers may evolve without centralized control and what the resulting challenges are.

LG 4-2: Use the "outside-in" approach for API design

One of the most important goals of APIs is to efficiently connect consumers with the provider's interface. API design should be focused on the needs of the consumers. It shouldn't be based on existing systems, existing use cases, or databases. Participants know and understand the approaches of API First and consumer-oriented API design.

LG 4-3: Understand openness and extensibility as important aspects for API evolution

Participants understand loose coupling, Postel's law, the concepts of forward and backward compatibility, and the importance of these concepts for the evolution of APIs. Participants also understand the consequences for API implementation in strong and statically typed languages.

LG 4-4: Understand API versioning

Participants gain knowledge of various aspects of versioning and how these are applied during the lifecycle of an API product. The following concepts are known:

- Compatibility and the difference between forward and backward compatibility
- Version identification in general and semantic versioning as a specific model for version identification
- Different ways of version identification for APIs (for example, domain name, URI path, HTTP header, payload-based)

LG 4-5: Know good practices for API design

Participants learn about established and widely used aspects for API design and understand their advantages and challenges. These aspects include:

- URL paths
- Using correct HTTP methods or operations for frequently needed functionality such as searching, filtering, or sorting
- Caching as a way to improve API performance
- Proposed formats such as JSON:API or HAL
- Problem Details for HTTP APIs (RFC 9457)

- API Design Style Guides

4.3. References

[Amundsen 2020], [Geewax 2021], [Higginbotham 2021], [HTTP Problem Details], [JSON], [Lauret 2019], [Zimmermann et al. 2022]

5. Description of APIs

Duration: 90 min	Practice time: 30 min
------------------	-----------------------

5.1. Terms and Principles

API description, OpenAPI, Protocol Buffers, AsyncAPI, GraphQL, gRPC, IDL

5.2. Learning Goals

LG 5-1: Understand the significance of API descriptions

Participants understand the value of API descriptions and know the target audience of those descriptions. They also know the problem of "API Drift". Participants understand the relationship between design, implementation, versioning, description, and description versioning, and they understand how these concepts relate ideally and in real-world scenarios.

LG 5-2: Understand OpenAPI as a description language for HTTP APIs

Participants understand OpenAPI as a description language that is specialized for HTTP-based APIs. They understand the history of OpenAPI and how the various versions evolved. Participants understand how to use OpenAPI for code generation both on the provider side and on the consumer side, and that this practice has some risks. They understand the overall structure of JSON and YAML.

LG 5-3: Understand OpenAPI's structure

Participants understand the main elements of an OpenAPI description and how to use them to describe APIs.

- Resources (Paths)
- Interactions (Operations)
- Representations
- Mechanisms such as Tags, Security, Components, and Webhooks

LG 5-4: Understand alternative description languages

OpenAPI is specialized for HTTP-based APIs. For other API styles, it is possible to use alternative descriptions languages that have the same general goals. Participants know these alternative description languages and can associate them with API styles.

- gRPC for RPC-oriented APIs
- AsyncAPI for event-oriented APIs
- GraphQL for query-oriented APIs
- The concept of an Interface Definition Language (IDL)

5.3. References

[\[AsyncAPI Specification\]](#), [\[GraphQL Specification\]](#), [\[gRPC Specification\]](#), [\[OpenAPI Specification\]](#), [\[Ponelat and Rosenstock 2022\]](#)

6. API Lifecycle and API Tooling

Duration: 60 min	Practice time: 30 min
------------------	-----------------------

6.1. Terms and Principles

API lifecycle, API product, linting, contract testing, API gateway

6.2. Learning Goals

LG 6-1: Understand the API lifecycle

Participants understand the various stages of the development cycle of an API product and the typical tasks that are performed throughout these stages. Participants understand the various stages of planning, requirements analysis, design, prototyping, development, testing, quality assurance, deployment, publication, operations, and maintenance, and how to improve an API product in an iterative way. Participants also know the different lifecycle stages of an API such as prototype, production, deprecation, and sunseting.

LG 6-2: Manage APIs as products

APIs are often used in loosely coupled scenarios and for this reason it makes sense to manage them as products. Participants understand what it means to manage an API as a product. API product management starts with focus on a set of consumers, deals with questions of usability, includes feedback, and also handles questions of deprecation and how to provide alternatives.

LG 6-3: Know API lifecycle tooling

Participants know typical tools that are used throughout the API lifecycle. These tools support producers as well as consumers and include functionality such as linting, testing (e.g., contract testing), mocking, and operations (API gateways). Participants gain practical knowledge of some of these tools and understand how they fit into the bigger picture of API lifecycle tooling.

6.3. References

[\[Nwaiwu 2024\]](#)

7. API Security

Duration: 60 min	Practice time: 30 min
------------------	-----------------------

7.1. Terms and Principles

Communication security, TCP, HTTP, SSL/TLS, HTTPS, HTTP authentication, OAuth, OpenID Connect

7.2. Learning Goals

LG 7-1: Understand communications security fundamentals

Participants understand the foundations of communications security and are able to relate them to technologies such as TCP, HTTP, and TLS. Participants understand the relevance of secure communications, even in the case of internal interfaces.

LG 7-2: Understand security technologies for APIs

Participants understand HTTPS, HTTP authentication, OAuth, and OpenID Connect, how they relate and differ, and how these technologies help when it comes to securing APIs.

LG 7-3: Understand OWASP API Security Top 10

Participants know the OWASP API Security Top 10 and understand the most common security issues of APIs.

7.3. References

[\[Ball 2022\]](#), [\[Madden 2020\]](#), [\[OWASP API Security Top 10\]](#)

8. APIs at Scale: Platforms and Governance

Duration: 60 min	Practice time: 0 min
------------------	----------------------

8.1. Terms and Principles

Internal developer platform (IDP), API platform, business platform, API guidelines, API design platform, self-service

8.2. Learning Goals

LG 8-1: Compare various notions of platform

Participants understand the various areas in which notions of platform are used. They understand how these notions are different and how they complement each other.

- Internal developer platform (IDP)
- API platform
- Business platform

Participants know how to differentiate between the mostly inward focus of an IDP and the both inwards and outwards directed focus of an API platform.

LG 8-2: Understand API guidelines

Participants understand the motivation for API guidelines. They understand the goals of creating harmonized practices for both API design and the implementation practices for APIs. Tools for supporting API guidelines such as linting are known and participants understand how they work.

LG 8-3: Establish and manage API guidelines

Participants know some examples of API guidelines by specific organizations. They also know good practices for managing and evolving API guidelines. Participants understand API guidelines as a living document that is managed in participatory ways which document actual practices.

LG 8-4: Use APIs as team interfaces

Participants know Team Topologies as an organizational model for making teams work more effective. The specific focus is on understanding how APIs are necessary to make the Team Topologies model work. These areas are the consumption of existing services (X-a-a-S model) and the way how platform teams make their platform available in a self-service model.

8.3. References

[Fishman and McLarty 2024], [Hohpe 2024], [Medjaoui et al. 2019]

References

This section contains references that are cited in the curriculum.

A

- [Amundsen 2020] Mike Amundsen, "Design and Build Great Web APIs: Robust, Reliable, and Resilient", Pragmatic, 2020
- [API Business Models] "ProgrammableWeb's 2020 Guide to API Business Models", ProgrammableWeb, May 2020. https://www.mulesoft.com/sites/default/files/cmm_files/2020_Guide_to_API_Business_Models.pdf
- [AsyncAPI Specification] AsyncAPI Specification. <https://www.asyncapi.com/docs/reference>

B

- [Ball 2022] Corey Ball, "Hacking APIs: Breaking Web Application Programming Interfaces", No Starch Press, 2022

F

- [Fishman and McLarty 2024] Stephen Fishman and Matt McLarty, "Unbundling the Enterprise: APIs, Optionality, and the Science of Happy Accidents", IT Revolution, 2024

G

- [Geewax 2021] JJ Geewax, "API Design Patterns", Manning Publications, 2021
- [Goyal 2023] Deepa Goyal, "API Analytics for Product Managers", Packt Publishing, 2023
- [GraphQL Specification] GraphQL Specification. <https://spec.graphql.org/>
- [gRPC Specification] gRPC Specification. <https://grpc.io/docs/>

H

- [Higginbotham 2021] James Higginbotham, "Principles of Web API Design: Delivering Value with APIs and Microservices", Addison-Wesley, 2021
- [Hohpe 2024] Gregor Hohpe, "Platform Strategy: Innovation Through Harmonization", 2024
- [Hohpe and Woolf 2003] Gregor Hohpe and Bobby Woolf, "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", Addison-Wesley, October 2003
- [HTTP Problem Details] Mark Nottingham, Erik Wilde, and Sanjay Dalal, "Problem Details for HTTP APIs", Internet Proposed Standard RFC 9457, July 2023

J

- [JSON] Tim Bray, "The JavaScript Object Notation (JSON) Data Interchange Format", Internet Standard RFC 8259, December 2017

L

- [Lauret 2019] Arnaud Lauret, "The Design of Web APIs", Manning Publications, 2019

M

- [Medjaoui et al. 2019] Mehdi Medjaoui, Erik Wilde, Ronnie Mitra, and Mike Amundsen, "Continuous API Management", O'Reilly, 2nd Edition, October 2021
- [Madden 2020] Neil Madden, "API Security in Action", Manning Publications, 2020

N

- [Nwaiwu 2024] Ikenna Nwaiwu, "Automating API Delivery: APIOps with OpenAPI", Manning Publications, July 2024

O

- [OpenAPI Specification] OpenAPI Specification. <https://spec.openapis.org/oas/latest.html>
- [OWASP API Security Top 10] "OWASP API Security Project", Open Worldwide Application Security Project. <https://owasp.org/www-project-api-security/>

P

- [Ponelat and Rosenstock 2022] Joshua S. Ponelat and Lukas L. Rosenstock, "Designing APIs with Swagger and OpenAPI", Manning Publications, 2022
- [Porcello 2018] Eve Porcello, "Learning GraphQL: Declarative Data Fetching for Modern Web Apps", O'Reilly, 2018

R

- [Richardson et al. 2013] Leonard Richardson, "RESTful Web APIs: Services for a Changing World", O'Reilly, 2013

Z

- [Zimmermann et al. 2022] Olaf Zimmermann, Mirko Stocker, Daniel Lübke, Uwe Zdun, Cesare Pautasso, "Patterns for API Design: Simplifying Integration with Loosely Coupled Message Exchanges", Addison-Wesley Professional, November 2022, ISBN 978-0-13767-010-9