# S4D437

## Transactional Apps with the ABAP RESTful Application Programming Model (RAP)

**COURSE OUTLINE**

Course Version: 24
Course Duration:

# SAP Copyrights, Trademarks and Disclaimers

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials may have been machine translated and may contain grammatical errors or inaccuracies.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

# Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

| | |
|---|---|
| This information is displayed in the instructor's presentation | 🖥 |
| Demonstration | ➤ |
| Procedure | 1<br>2<br>3 |
| Warning or Caution | ⚠ |
| Hint | 💡 |
| Related or Additional Information | » |
| Facilitated Discussion | 💬 |
| User interface control | *Example text* |
| Window title | *Example text* |

# Contents

# Course Overview

## TARGET AUDIENCE

This course is intended for the following audiences:

- Development Consultant
- Developer

# UNIT 1 Exploring the ABAP RESTful Application Programming Model (RAP)

## Lesson 1: Exploring the Concept and Architecture of RAP

### Lesson Objectives

After completing this lesson, you will be able to:

- Understand the concept.
- Use ABAP development tools.
- Explain the architecture.

## Lesson 2: Defining an OData UI Service

### Lesson Objectives

After completing this lesson, you will be able to:

- Define a CDS projection view.
- Enrich a projection view with UI metadata.
- Create and preview an OData UI service.

# UNIT 2 Working with Business Objects

## Lesson 1: Defining Business Objects and their Behavior

### Lesson Objectives

After completing this lesson, you will be able to:

- Create a CDS behavior definition.
- Generate a behavior implementation class.
- Create a CDS behavior projection.

## Lesson 2: Using the Entity Manipulation Language (EML) to Access Business Objects

### Lesson Objectives

After completing this lesson, you will be able to:

- Describe the purpose and syntax of EML.
- Describe the derived data types for RAP Business Objects.
- Use the Entity Manipulation Language (EML).

## Lesson 3: Understanding Concurrency Control

### Lesson Objectives

After completing this lesson, you will be able to:

- Describe pessimistic concurrency control (locking).
- Enable optimistic concurrency control.

## Lesson 4: Defining Actions and Messages

### Lesson Objectives

After completing this lesson, you will be able to:

- Define and implement an action.
- Expose an action to an OData service.
- Define a button in SAP Fiori elements.

- Access application data in behavior implementations.
- Define and return messages.

## Lesson 5: Implementing Authority Checks

### Lesson Objectives

After completing this lesson, you will be able to:

- Restrict read access with access controls.
- Implement explicit authority checks.

# UNIT 3 Adding Basic Operations Update and Create

## Lesson 1: Defining Static Field Control and Value Helps

### Lesson Objectives

After completing this lesson, you will be able to:

- Enable editing.
- Set input fields to read-only and mandatory.
- Define value help for input fields.

## Lesson 2: Implementing Input Checks Using Validations

### Lesson Objectives

After completing this lesson, you will be able to:

- Explain validations.
- Define and implement input checks.
- Link messages to input fields.

## Lesson 3: Setting Values Using Numbering and Determinations

### Lesson Objectives

After completing this lesson, you will be able to:

- Describe the numbering concepts.
- Define and implement determinations.

## Lesson 4: Implementing Dynamic Feature Control

### Lesson Objectives

After completing this lesson, you will be able to:

- Explain dynamic action, operation, and field control in RAP.
- Implement dynamic feature control.

# UNIT 4 Developing Draft-Enabled Services

## Lesson 1: Understanding the Draft Concept

### Lesson Objectives

After completing this lesson, you will be able to:

- Explain the need for draft in stateless applications.
- Enable draft handling in the Business Object.

## Lesson 2: Implementing the Behavior of Draft-Enabled BOs

### Lesson Objectives

After completing this lesson, you will be able to:

- Enable draft handling in an OData UI service.
- Explain the difference between transition messages and state messages.
- Describe the draft-specifics in behavior implementations.

## Lesson 3: Defining Determine Actions and Side Effect

### Lesson Objectives

After completing this lesson, you will be able to:

- Define determine actions.
- React on user input using side effects.

# UNIT 5 Defining Compositions

## Lesson 1: Defining Composite Business Objects

### Lesson Objectives

After completing this lesson, you will be able to:

- Define compositions.

## Lesson 2: Defining Compositions in OData UI Services

### Lesson Objectives

After completing this lesson, you will be able to:

- Expose compositions to OData services.
- Enable navigation in SAP Fiori elements apps.

## Lesson 3: Implementing the Behavior of Composite Business Objects

### Lesson Objectives

After completing this lesson, you will be able to:

- Access composite business objects using EML.

# UNIT 6 Implementing Unmanaged Data Access

## Lesson 1: Exploring the Behavior Implementation Options

### Lesson Objectives

After completing this lesson, you will be able to:

- Describe the data flow in the Business Object Runtime.
- Describe the behavior implementation options.

## Lesson 2: Implementing Unmanaged Business Objects

### Lesson Objectives

After completing this lesson, you will be able to:

- Implement an unmanaged business object.

## Lesson 3: Implementing Data Access in Managed Business Objects

### Lesson Objectives

After completing this lesson, you will be able to:

- Implement unmanaged save and additional save.

# UNIT 7 Integrating Business Events

## Lesson 1: Raising and Handling Business Events

### Lesson Objectives

After completing this lesson, you will be able to:

- Describe Business Events.
- Define and raise business events in RAP.
- Handle business events locally.

## Lesson 2: Working with Event Parameters

### Lesson Objectives

After completing this lesson, you will be able to:

- Define CDS Abstract Entities.
- Raise business events with parameters.
- Handle business events with parameters.

# UNIT 8 Enabling and Using Extensibility

## Lesson 1: Exploring Extensibility Options

### Lesson Objectives

After completing this lesson, you will be able to:

- Name the extension scenarios in ABAP Cloud.
- Describe developer extensibility in RAP.

## Lesson 2: Enabling Data Model Extensibility

### Lesson Objectives

After completing this lesson, you will be able to:

- Enable database tables for extension.
- Enable CDS views for extension.

## Lesson 3: Developing Data Model Extensions

### Lesson Objectives

After completing this lesson, you will be able to:

- Extend database tables.
- Extend CDS views.

## Lesson 4: Enabling and Developing Behavior Extensions

### Lesson Objectives

After completing this lesson, you will be able to:

- Enable behavior extensions.
- Extend the behavior of a business object.