

In this video tutorial, I will discuss with you about the evolution towards the RESTful application

programming model.

Let's start.

Abap is a high level programming language created by SAP.

Over the years, the Java programming model has evolved significantly to meet the challenging business

needs and technological advancements.

Initially, we had the classical Abap programming model, which worked up until Abap platform 7.5.

This programming model supported freestyle Abap programming.

This was followed by Abap programming model for SAP Fiori.

The Abap programming model for SAP Fiori words from Abap platform 7.5 onwards.

It allowed developers to create Fiori apps using CDs based annotations without much uf5 coding.

However, it required the use of Bopf business object processing framework for transactional application

processing.

That is, to perform, create, update and delete operations.

The drawback with Bopf is that it is not possible to catch errors during design time.

We get to know about the errors when we execute it, and it also becomes very difficult to identify

issues because of so many autogenerated goals.

The latest evolution in this journey is the Abap RESTful Application Programming model.

This model is a strategic long Tum solution for Abap development.

It provides an efficient way to build enterprise ready SAP Hana optimized OData based Fury, US services

and web APIs.

One of the key advantages of Wrap model is that it allows developers to focus more on business logic

than technical aspects.

For example, in this programming model, BPF is not required for transactional processing in fury applications.

Comparing to the previous programming models, it provides a programming model as a strategic long Tum

solution for Abap development.

It offers an end to end development experience with the standardised architecture and dataflow to support

high development efficiency.

In summary, the evolution towards wrapping the ACP Abap represents a significant step forward in terms

of efficiency, flexibility and modern programming paradigm.

That's it for now.

See you in the next video lecture.

Autoscroll

Course content

Overview

Q&AQuestions and answers

Notes

Announcements

Reviews

Learning tools