

O Fim do Dilema: Dominando o Flexible Programming Model

Como construir aplicativos SAP Fiori que são rápidos, estáveis
E flexíveis.

O Dilema Histórico no Desenvolvimento Fiori

Fiori Elements Padrão



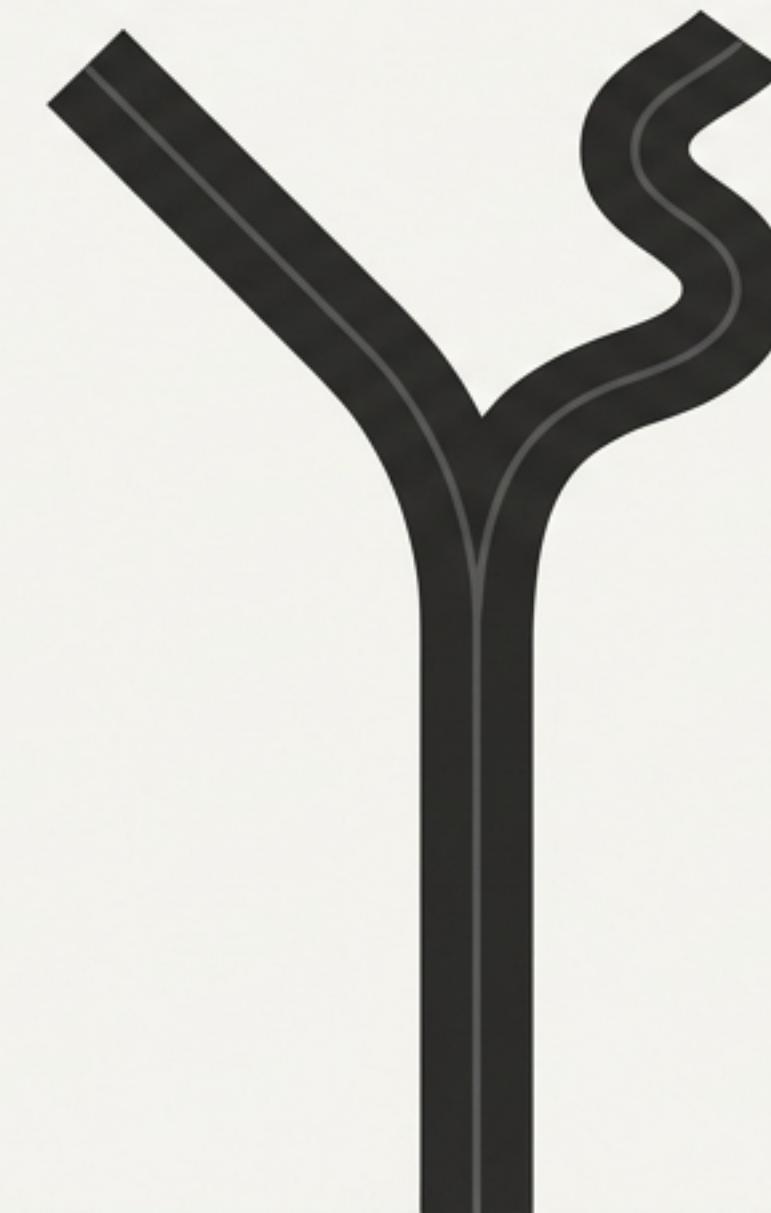
Vantagens

- Desenvolvimento extremamente rápido, baixo custo de manutenção, estabilidade garantida.

Desvantagem

- Uma "caixa preta". Requisitos fora do padrão eram impossíveis ou exigiam "hacks" perigosos que quebravam em atualizações.

Padrão



Flexível



Freestyle UI5

Vantagens

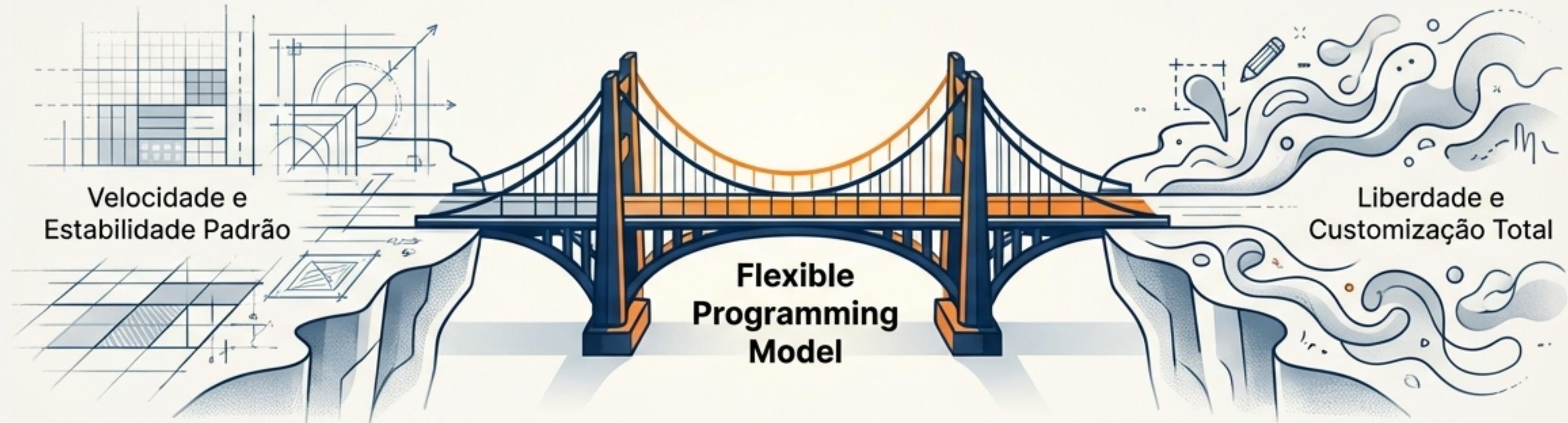
- Flexibilidade total, controle sobre cada pixel.

Desvantagem

- Desenvolvimento lento, alto Custo Total de Propriedade (TCO), manutenções complexas em atualizações de design system.

A Solução: O Melhor dos Dois Mundos com FPM

O Flexible Programming Model (FPM), introduzido com OData V4, permite que você mantenha a estabilidade do padrão e injete customizações de forma suportada e inteligente.



90%

Mantenha a maior parte
do aplicativo como
Fiori Elements padrão
(rápido, barato, estável).



10%

Injete código
customizado apenas
apenas onde
realmente precisa de
algo especial.



Resultado: Você não
“quebra” o padrão,
você o “estende”.

A Inovação Central: Building Blocks (O 'Lego' da SAP)

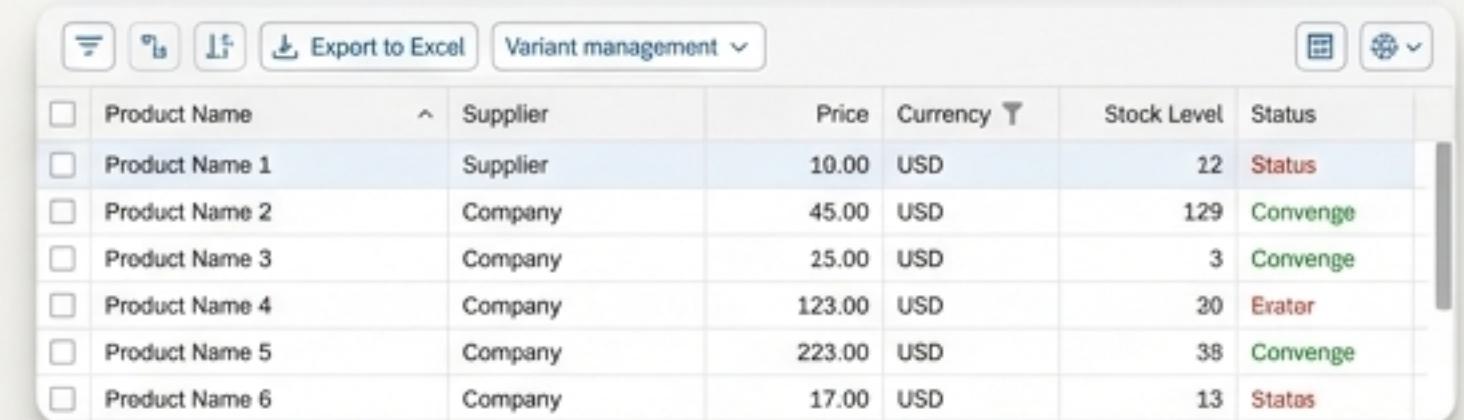
Building Blocks são 'tags mágicas' (namespace `sap.fe.macros`) que trazem a inteligência do Fiori Elements para dentro de suas views customizadas.

A Simplicidade do XML

```
<!-- Dentro da sua View XML Customizada -->  
<macros:Table  
    metaPath="@com.sap.vocabularies.UI.v1.LineItem"  
    id="MySmartTable" />
```



O Poder Gerado



	Product Name	Supplier	Price	Currency	Stock Level	Status
<input type="checkbox"/>	Product Name 1	Supplier	10.00	USD	22	Status
<input type="checkbox"/>	Product Name 2	Company	45.00	USD	129	Convene
<input type="checkbox"/>	Product Name 3	Company	25.00	USD	3	Convene
<input type="checkbox"/>	Product Name 4	Company	123.00	USD	20	Erater
<input type="checkbox"/>	Product Name 5	Company	223.00	USD	38	Convene
<input type="checkbox"/>	Product Name 6	Company	17.00	USD	13	Status

- ✓ Tabela Fiori Elements completa
- ✓ Conectada ao serviço OData
- ✓ Barra de ferramentas com exportação para Excel
- ✓ Variantes e personalização de colunas
- ✓ Tudo lendo as anotações (@UI.lineItem) do backend.

Um Kit de Ferramentas Inteligente e Reutilizável

Além de tabelas, o FPM oferece um conjunto de componentes prontos para acelerar o desenvolvimento de seções customizadas.



<macros:FilterBar>

Renderiza uma barra de filtros inteligente e adaptativa, baseada nas anotações de `SelectionFields`.



<macros:Field>

Cria um campo de formulário completo, com label, F4 (Value Help) e validação automática.

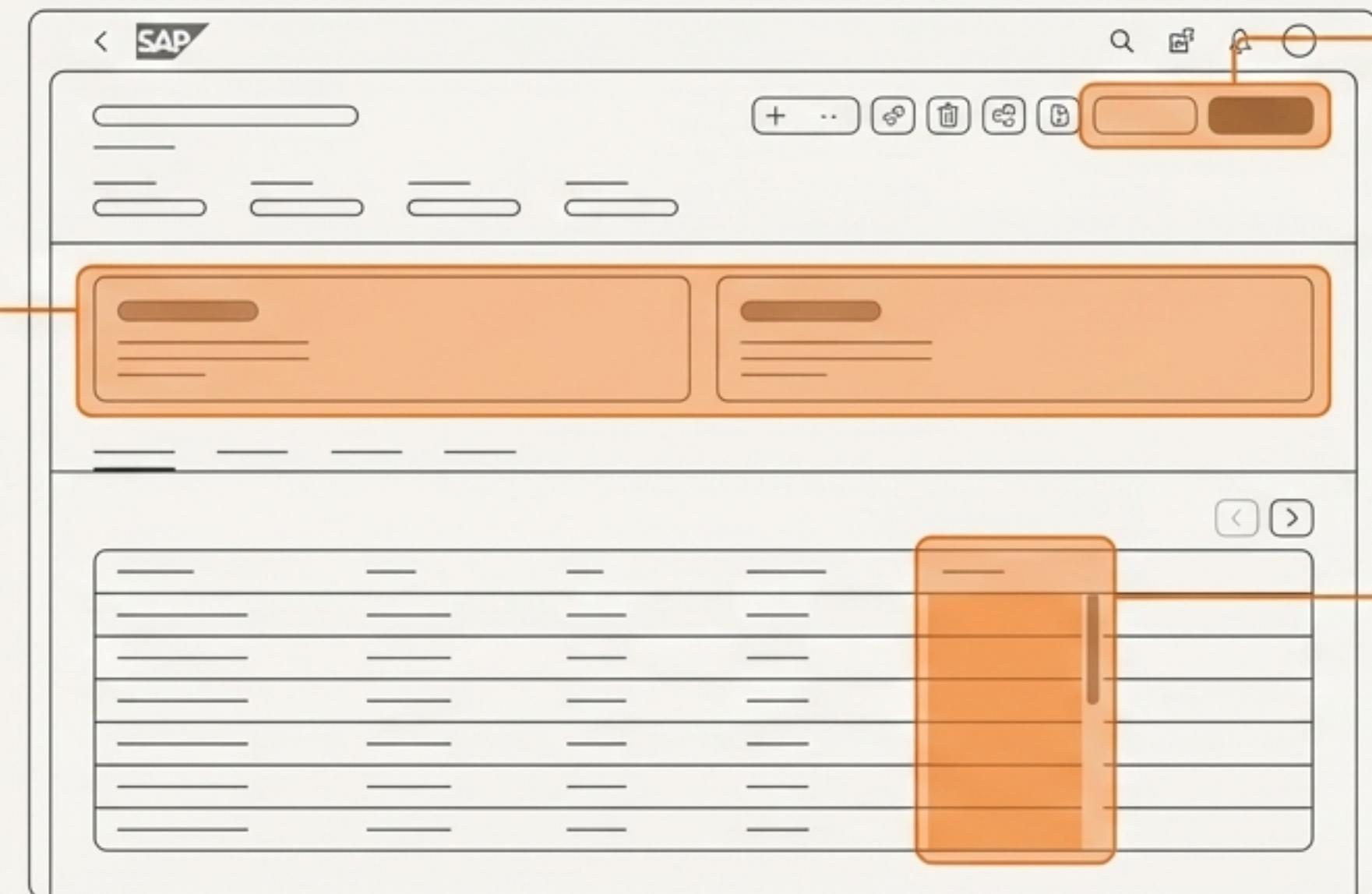


<macros:Chart>

Gera gráficos analíticos interativos baseados nas anotações `@UI.Chart`.

Onde Injetar Customizações: Entendendo os Extension Points

O FPM oferece pontos de inserção precisos e seguros na estrutura padrão do aplicativo. Isso garante que suas customizações sobrevivam a upgrades futuros.



Custom Sections

Adicione blocos de conteúdo ou abas inteiras.

Custom Actions

Adicione botões com lógica customizada na barra de ferramentas.

Custom Columns

Insira controles complexos dentro de colunas de tabelas.

O Que Você Pode Estender?



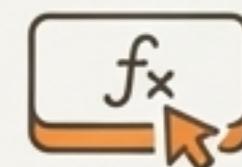
Custom Sections (Seções Customizadas)

Técnica

Fragmento ou View XML registrado no `manifest.json`.

Caso de Uso

Incorporar mapas (Google Maps), gráficos de bibliotecas de terceiros (D3.js), players de vídeo ou exibir dados de múltiplas fontes (Mashups).



Custom Actions (Ações Customizadas)

Técnica

Registro no `manifest.json` apontando para um método na Controller Extension.

Caso de Uso

Abrir um PDF, chamar uma API REST externa (Consulta de CEP), abrir um diálogo (Popup) complexo, navegação customizada.



Custom Columns (Colunas Customizadas)

Técnica

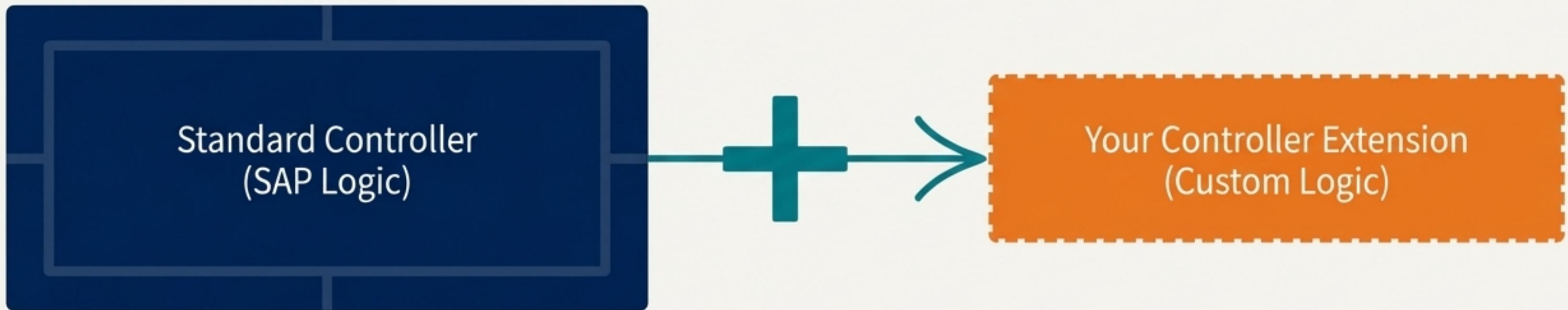
Fragmento XML injetado na definição da tabela.

Caso de Uso

Exibir Micro Charts (Sparklines), Sliders para ajuste de valores, botões de ação inline, formatação condicional complexa.

Injetando Lógica: O Papel das Controller Extensions

Quando adicionamos views ou fragmentos customizados, precisamos de lógica JavaScript. No FPM, não substituímos o controlador padrão da página; **nós o estendemos.**



- Você cria um arquivo .js que coexiste com a lógica padrão da SAP.
- Permite adicionar código para manipular eventos (cliques, mudanças de valor) de forma segura.
- Garante que sua lógica customizada não seja sobrescrita em atualizações do framework.

Controle Fino: Hooks de Ciclo de Vida e a Extension API



Hooks de Ciclo de Vida e Overrides

`override.onInit`: Execute código assim que a página é inicializada (ex: carregar dados de APIs externas).

`override.editFlow`: Intercepte eventos transacionais. Injeta código `onBeforeSave` para validações extras ou formatação de dados.

`override.routing`: Controle a navegação e redirecione o usuário para rotas customizadas.



A Extension API: Sua Ponte Oficial

Acessada via `this.base.getExtensionAPI()`, é a forma segura e suportada de interagir com a página padrão.

- Recarregar a tabela (`refresh()`).
- Obter os contextos selecionados (`getSelectedContexts()`).
- Navegar para outras páginas.
- Gerenciar mensagens de erro.

Como Implementar: O Fluxo de Trabalho Visual com Fiori Tools

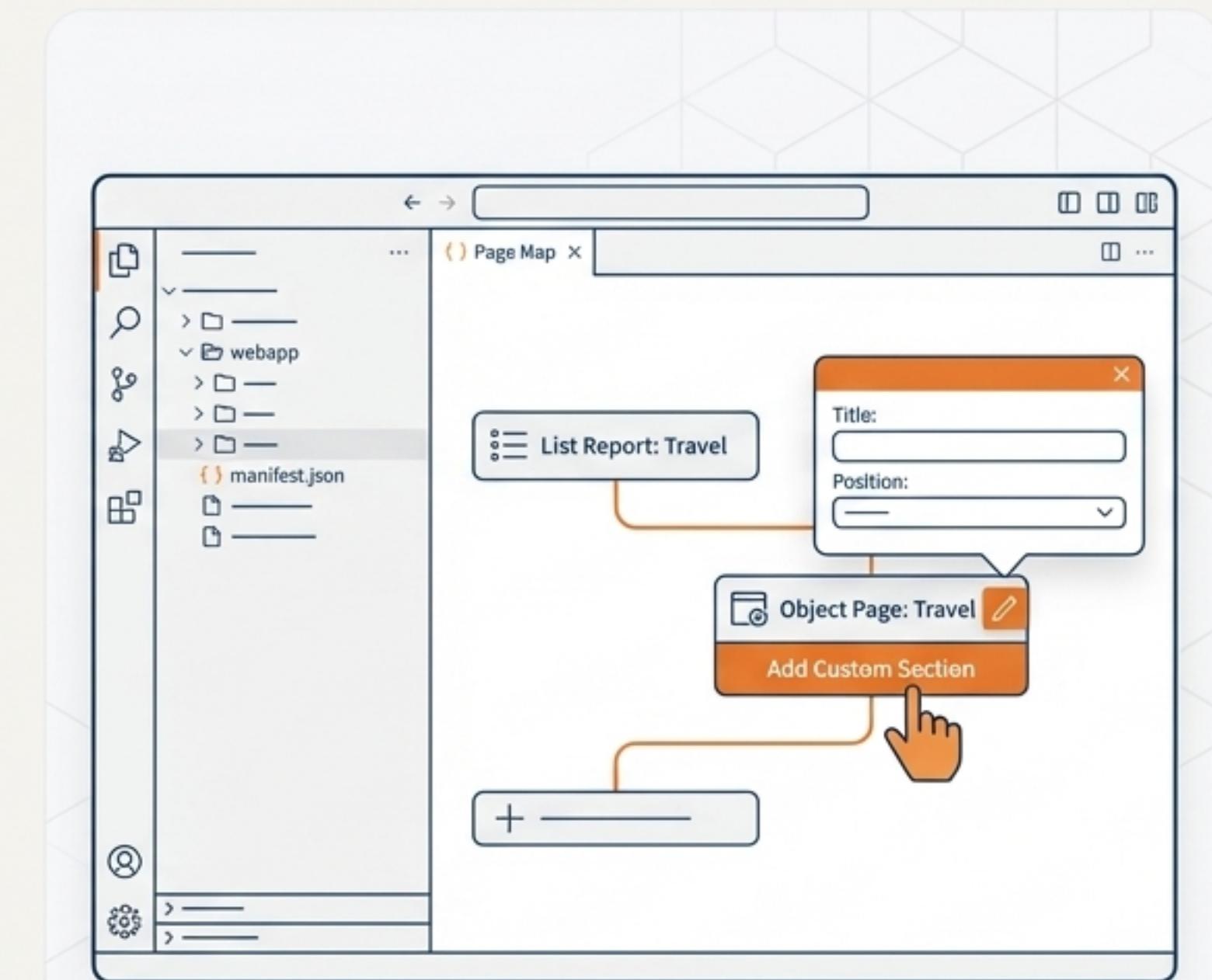
Embora seja possível editar o manifest.json manualmente, o uso do Page Map é a prática recomendada para evitar erros de sintaxe.

Fluxo de Trabalho Simplificado

- 1. Abra o Projeto:** No VS Code / BAS.
- 2. Acesse o Page Map:** Clique com o botão direito em webapp > Show Page Map.
- 3. Selecione a Página:** Ex: Object Page da entidade Travel.
- 4. Adicione a Extensão:** Clique no ícone de lápis ou em Add Custom Section.
- 5. Configure:** Defina o título e a posição da seção.

O Que a Ferramenta Faz por Você

- ✓ Gera o arquivo .fragment.xml.
- ✓ Cria o arquivo .js da Controller Extension.
- ✓ Adiciona automaticamente as configurações complexas no manifest.json.



Colocando Tudo Junto: Anatomia de uma Página FPM

Padrão (Gerado pelo RAP)

Padrão (Gerado via Anotações)

CUSTOM - Injetado via FPM

Sua View XML com HTML Container para Google Maps API

<macros:Field> (Building Block lendo do OData)

Padrão (Tabela 'LineItem' do RAP)

Travel 1024: Business Trip to London

Status: Imminentenon Start Date: ACME Corp. End Date: Jan 29, 2024 Billed verstand: 275.796 Complete Prelaries: -

General Information

Travel ID: 1024 Customer: ACME Corp. Start Date: Jan 15, 2024 End Date: Jan 20, 2024 Grantentoate:

Geolocalização

Map Satellite Latitude: 51.5074° N Longitude: 0.1278° W

Booking Items

Item No.	Booking ID	Airline	Flight No.	Price	Total
Item 01	002A000233	ACME Corp.	P8550	136.00	120.00
Item 02	003A000234	Airline	P8054	204.00	116.00
Item 03	003A000235	Airline	Flight No.	467.00	135.00
Item 04	0030000266	ACME Corp.	P4066	125.00	124.00
Item 05	003A000260	Airline	P8323	232.00	123.00
Item 06	0030000066	Airline	P3024	132.00	123.00
Item 07	003A000228	ACME Corp.	P8085	165.00	120.00

Quando Usar FPM? Um Guia Prático de Decisão

Cenário	Solução Recomendada	Motivo
CRUD simples de uma tabela padrão	Fiori Elements Padrão (RAP)	Mais rápido de desenvolver, zero código de frontend, manutenção mínima.
Tabela padrão com uma coluna de botão extra que chama uma API externa	Fiori Elements + Custom Action / Custom Column	Mantém 95% do padrão, adiciona apenas a lógica de negócio necessária.
Página de detalhes com layout exótico (ex: mapa interativo com dados)	Fiori Elements + Custom Section	Isola a complexidade customizada em uma seção, mantendo o restante da página padrão e estável.
Dashboard analítico complexo com interações e visualizações não-padrão	Freestyle UI5 (ou FPM com uso intensivo)	O esforço para adaptar o padrão seria maior do que construir do zero.

Glossário Essencial do FPM

Flexible Programming Model (FPM)

Arquitetura para OData V4 que permite estender apps Fiori Elements com componentes customizados (UI5) de forma suportada e estável.

Building Blocks (Macros)

Componentes de UI (sap.fe.macros) que encapsulam funcionalidades complexas do Fiori Elements (tabelas, filtros) para uso em views customizadas.

Extension Point

Locais pré-definidos e seguros na estrutura da página (seções, tabelas, barras de ferramentas) para inserir conteúdo customizado.

Controller Extension

Arquivo JavaScript que estende o controlador padrão, permitindo adicionar lógica de frontend customizada e interagir com a página via ExtensionAPI.

ExtensionAPI

Interface pública fornecida pelo Fiori Elements para que o código customizado interaja com o aplicativo padrão de forma segura (ex: refresh(), getSelectedContexts()).