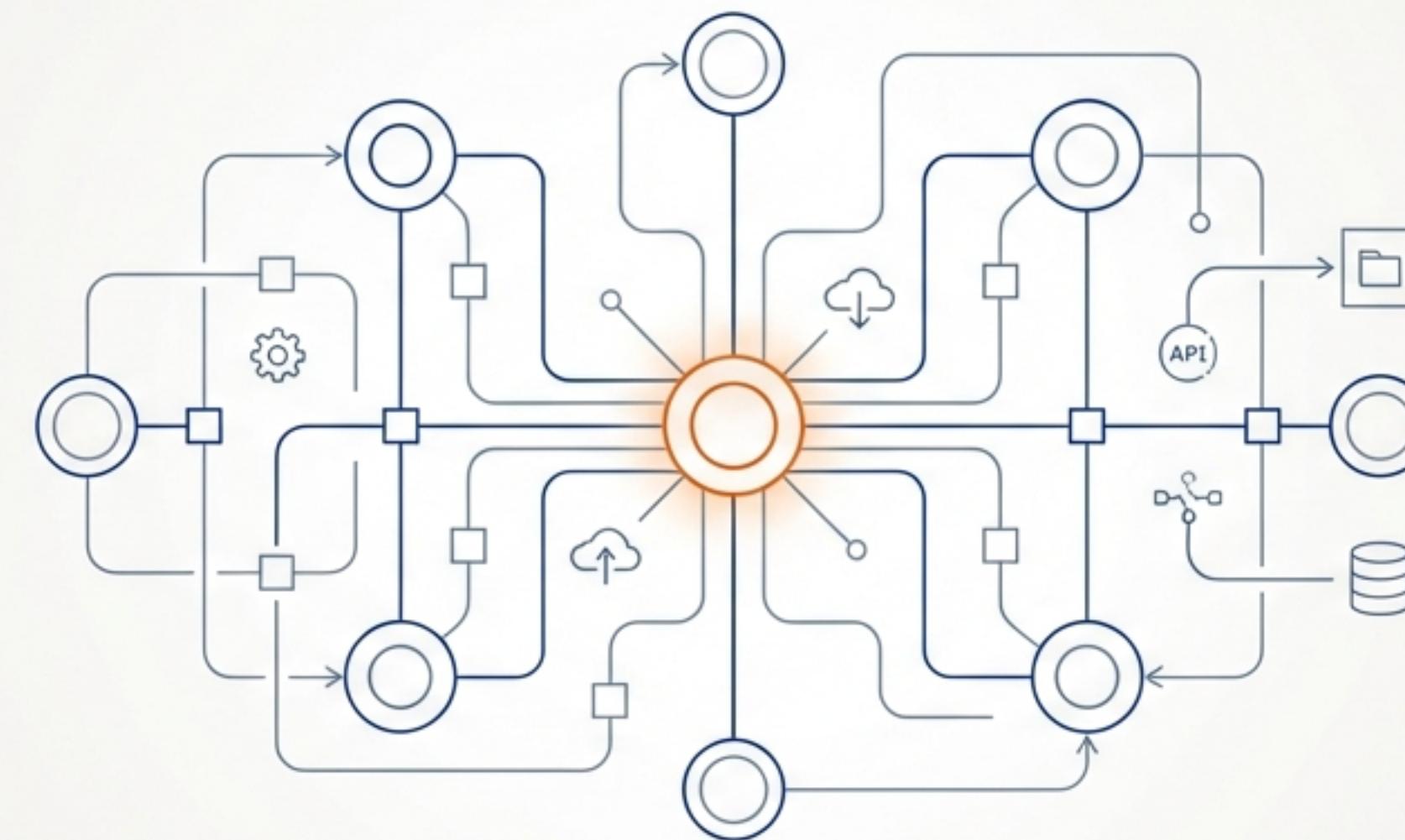


Dominando a Integração de Serviços Remotos

Um guia visual para configurar `cds.RemoteService` no CAP para Node.js.



SAP Cloud Application Programming Model

A ponte para seus serviços externos.

No ecossistema CAP, `cds.RemoteService` é a classe proxy fundamental para consumir serviços remotos. Ela simplifica a comunicação com APIs externas, seja via OData ou REST.



Class `cds.RemoteService` is a service proxy class to consume remote services via different protocols, like OData or plain REST.

A jornada da configuração: da segurança à confiabilidade.

Dominar `cds.RemoteService` significa ir além do básico e configurar dois aspectos críticos para aplicações robustas:



Protegendo as Conexões

Implementando o tratamento de CSRF-Token para garantir que as requisições sejam seguras e autênticas.



Garantindo a Resposta

Gerenciando Timeouts para evitar que sua aplicação fique presa aguardando por serviços lentos.

Capítulo 1: Protegendo as Conexões com CSRF

Ativando a proteção CSRF de forma simples.

Para sistemas remotos que exigem proteção contra Cross-Site Request Forgery, você pode habilitar o tratamento de token CSRF nativo do @sap-cloud-sdk/core diretamente na sua configuração.

As opções `csrf: true` e `csrfInBatch: true` são presets convenientes para habilitar a proteção em requisições regulares e em operações em lote (batch).

Seu serviço remoto

```
"cds": {  
  "requires": {  
    "API_BUSINESS_PARTNER": {  
      "kind": "odata",  
      "model": "srv/external/API_BUSINESS_PARTNER",  
  
      "csrf": true, —————— Habilita CSRF para  
      "csrfInBatch": true —————— requisições padrão  
    }  
  }  
}
```

Habilita CSRF para requisições padrão

Habilita CSRF para operações em lote

Controle avançado sobre o tratamento de CSRF.

Quando a configuração padrão não é suficiente, você pode customizar o comportamento da busca pelo token CSRF fornecendo um objeto de configuração detalhado.

Parâmetro: method

Define o método HTTP para buscar o token.

Padrão: \ "head\ "

"API_BUSINESS_PARTNER": {

...

```
  "csrf": {  
    "method": "get",  
    "url": "..."  
  }
```

}

...

URL customizada para
o fetch do token

Parâmetro: url

A URL específica para buscar o token.

Padrão: O caminho do recurso da requisição,
sem parâmetros.



Nota do Especialista

Use esta configuração avançada para se integrar com serviços que possuem implementações de CSRF não-padrão.

Capítulo 2: Garantindo a Resposta com Timeouts

Evitando longas esperas com `requestTimeout`.

Uma aplicação resiliente não pode esperar indefinidamente por uma resposta de um serviço externo. Isso pode levar a threads bloqueadas e a uma má experiência do usuário.

A configuração `requestTimeout` define o tempo máximo, em milissegundos, que o CAP aguardará por uma resposta antes de encerrar a requisição.

Valor Padrão: 60000 ms (60 segundos)



```
"API_BUSINESS_PARTNER": {  
    "kind": "odata",  
    ...  
    "credentials": { ... },  
  
    "requestTimeout": 100000  
}
```

Define o timeout para 100 segundos

Resumo das Configurações Essenciais

Segurança CSRF (Básico)



Propriedades

```
'csrf: true',  
'csrfInBatch: true'
```

Uso

Habilitação rápida da proteção CSRF para requisições padrão e em lote.

Segurança CSRF (Avançado)



Propriedade

```
'csrf: { ... }'
```

Parâmetros

```
'method', 'url'
```

Uso

Customização fina do processo de obtenção do token CSRF.

Confiabilidade (Timeout)



Propriedade

```
'requestTimeout: <ms>'
```

Uso

Definição do tempo máximo de espera por uma resposta do serviço remoto.

A jornada de integração está sempre evoluindo.

As funcionalidades de cds.RemoteService continuam a ser expandidas para cobrir mais cenários e aumentar a flexibilidade e a disponibilidade das integrações. O que você aprendeu hoje é a base sólida para construir aplicações robustas.

> “*There’s more to come in this place in upcoming releases.*”