



JAVA CAP: SEU CENTRO DE COMANDO DE DESENVOLVIMENTO

Domine a complexidade e acelere seu fluxo de trabalho com o novo
Developer Dashboard.

Uma Visão Centralizada para Gerenciar e Monitorar Suas Aplicações

O CAP Developer Dashboard simplifica o desenvolvimento ao fornecer um ponto centralizado onde os desenvolvedores podem gerenciar e monitorar suas aplicações CAP de forma eficiente. Ele oferece ferramentas e funções para apoiar o processo de desenvolvimento e ajuda os desenvolvedores a identificar e resolver problemas rapidamente.

The screenshot displays the CAP Developer Dashboard interface. At the top, there's a header with metrics: % SERVICES (600KB), % ENTITIES (500KB), % EVENTS (500KB), % EMITS (600T), and a timestamp (local 04/08/2024). Below the header is a line chart showing system status over time, with a red line fluctuating between 0 and 40. To the right of the chart is a bar chart titled 'CRUD Events' showing counts for Create (~5), Read (~48), Update (~2), and Delete (~2).

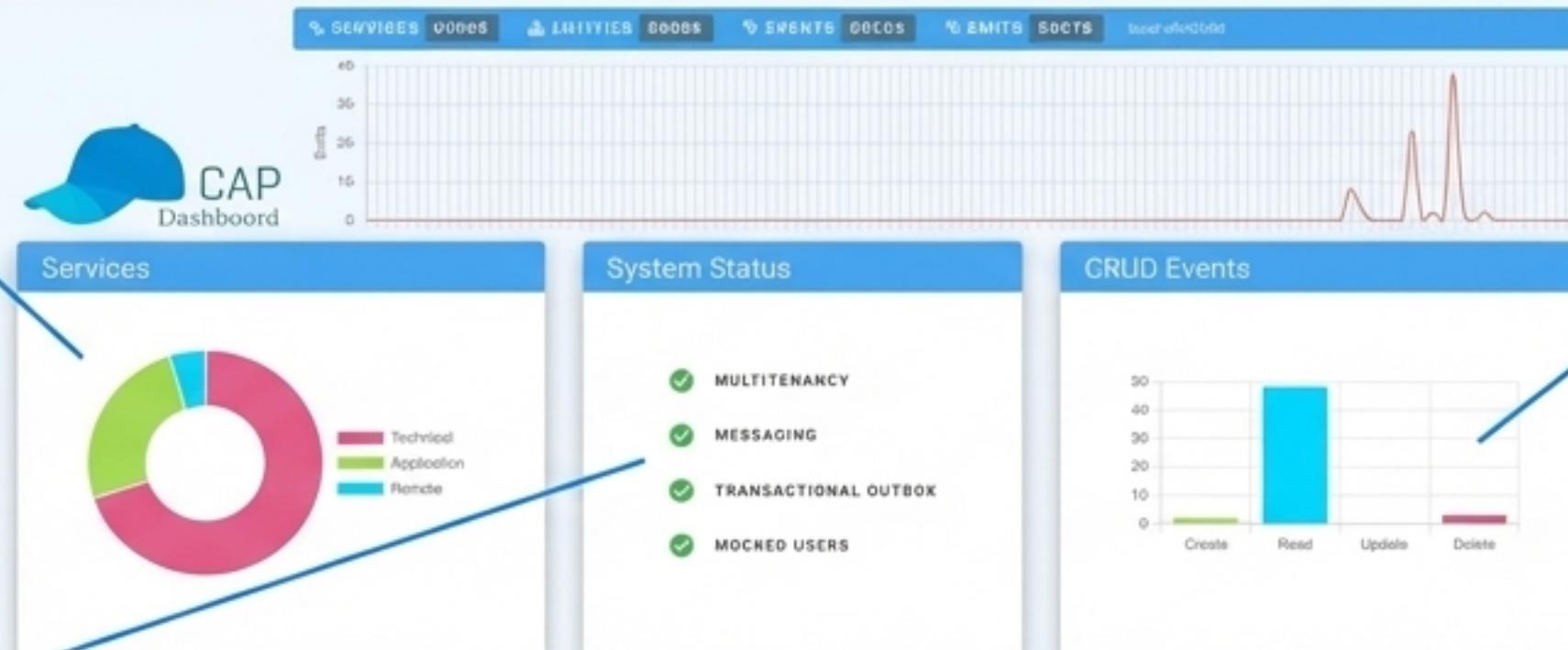
The main area is divided into three sections:

- Services:** A pie chart showing the distribution of services: Technical (pink), Application (green), and Remote (blue). The Application segment is the largest.
- System Status:** A list of checked items indicating system features:
 - MULTITENANCY
 - MESSAGING
 - TRANSACTIONAL OUTBOX
 - MOCKED USERS
- CRUD Events:** A bar chart showing the volume of CRUD operations: Create (~5), Read (~48), Update (~2), and Delete (~2).

At the bottom of the dashboard is a terminal window titled 'Console' showing log output. The logs include error messages such as 'Error executing the statement (service 'CustomerService', entity 'Customer', event 'Create', entity 'Customer_POCHOLDER_A', payload='customer'))' and 'Error committing the transaction: Error executing the statement (service 'CustomerService', entity 'Customer', event 'Create', entity 'Customer_POCHOLDER_A', payload='customer'))'. There are also several 'Info' and 'Warn' messages related to database connections and statements.

Explorando Seu Painel de Controle

Monitore o status e a atividade
de todos os seus serviços.



Verifique a integração de
componentes CAP como
multitenancy, messaging e
resilience.

A terminal-like window titled "Console" displays a series of log entries in a monospaced font. The logs include various database queries, error messages, and system notifications related to service operations like "Delete", "Update", and "Create". One prominent error message states: "Error executing the statement (operator 'DeleteServiceDefault', event '0009', entity 'DET_BESTDOSE_POTROR_A_PublicoEfetivaDose')".

Visualize a frequência
de operações de Create, Read,
Update e Delete em tempo real.

Acesse logs detalhados
diretamente na interface para
um debugging eficiente.



Primeiros Passos: Configuração Local

Vamos começar. Veja como habilitar o dashboard em seu ambiente de desenvolvimento local em apenas dois passos.

Passo 1: Adicione a Dependência ao seu Projeto



Para começar, adicione a feature `cds-feature-dev-dashboard` às suas dependências do Maven.

```
<dependency>
  <groupId>com.sap.cds</groupId>
  <artifactId>cds-feature-dev-
  dashboard</artifactId>
</dependency>
```

Passo 2: Garanta o Acesso Autorizado



Por padrão, o acesso ao dashboard é protegido e requer a role `cds.Developer`.

- O usuário mock padrão, `developer`, já possui essa role configurada.
- Se você utiliza usuários mock customizados, deve atribuir a role `cds.Developer` a eles.

Exemplo de Configuração (application.yaml):

```
cds:  
  security:  
    mock:  
      users:  
        - name: myUser  
          password: myPass  
          roles:  
            - cds.Developer
```





Subindo de Nível: Setup na Nuvem

Pronto para usar o dashboard em seu cenário de desenvolvimento na nuvem?
Vamos configurar um exemplo para BTP Cloud Foundry com Approuter e XSUAA.

Visão Geral da Configuração na Nuvem

Para habilitar o dashboard em um ambiente BTP, são necessários três ajustes principais em sua configuração:



Desativar o Perfil de Produção

Ajuste seu `mta.yaml` para garantir que o ambiente de desenvolvimento esteja ativo.



Adicionar a Role de Desenvolvedor

Atualize seu `xs-security.json` para incluir la role `cds.Developer`.



Configurar o Approuter

Habilite o suporte a WebSockets e defina as rotas do dashboard no `xs-app.json`.



A REGRA DE OURO: APENAS PARA DESENVOLVIMENTO

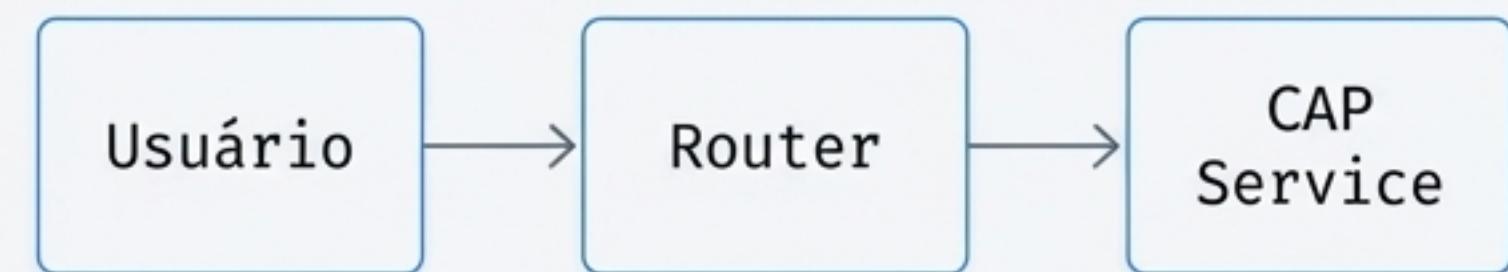
É estritamente proibido usar o dashboard em um ambiente de produção. Ele permite o acesso a dados sensíveis e representa um risco de segurança.

A role `cds.Developer` deve ser usada apenas com usuários de teste. É altamente recomendável não usar essa role com usuários que possam ser usados em sistemas de produção.

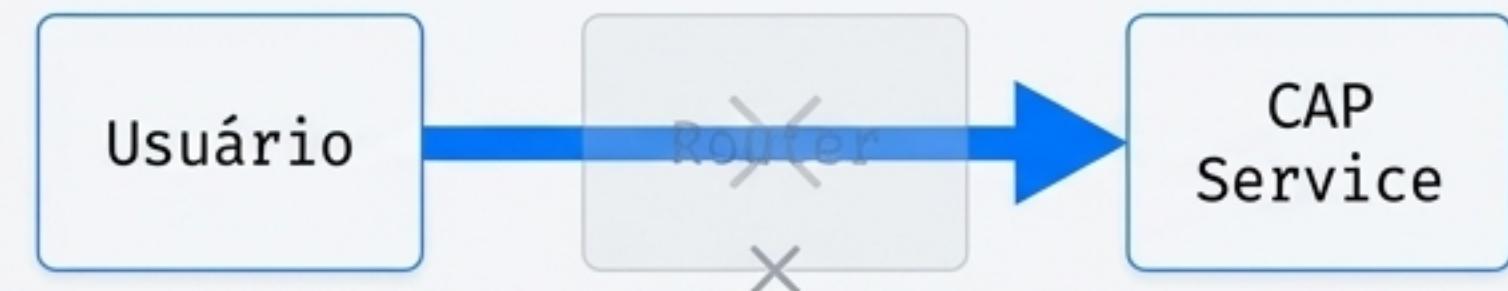
Cenários Avançados: Acesso Direto sem Roteador

Em alguns casos, sua aplicação pode rodar em um ambiente complexo e você pode precisar acessar o dashboard diretamente, sem um roteador. Para isso, é possível desabilitar a checagem de autorização.

Fluxo Padrão



Fluxo Avançado (Acesso Direto)



Opção 1: Desabilitar Autorização via Configuração

Você pode desabilitar a autorização usando uma das seguintes opções de configuração:

Em application.yaml

```
# Em application.yaml
cds:
  dashboard:
    authorization:
      enabled: false
```

Em mta.yaml (como variável de ambiente)

```
# Em mta.yaml
CDS_ENVIRONMENT_PRODUCTION_ENABLED: false
```

Opção 2: Desabilitar Autenticação via Código Java

Como alternativa, desabilite a autenticação para as rotas do dashboard diretamente em sua configuração do Spring Security.

```
import static org.springframework.security.web.util.matcher.AntPathRequestMatcher.antMatcher;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.annotation.Order;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
public class WebSecurity {
    @Bean
    @Order(1)
    public SecurityFilterChain appFilterChain(HttpSecurity http) throws Exception {
        return http
            .securityMatchers(m -> m.requestMatchers(antMatcher("/dashboard/**")))
            .authorizeHttpRequests(auth -> auth.anyRequest().permitAll())
            .csrf(c -> c.disable())
            .build();
    }
}
```

Seu Centro de Comando Está Pronto



O CAP Developer Dashboard é mais do que uma ferramenta; é um aliado essencial para trazer clareza, controle e eficiência ao seu ciclo de desenvolvimento.

Para mais informações, consulte a apresentação do Developer Dashboard na conferência RECAP 2024.