



Qlik Sense

Mashup & Extensions Overview

Ajay Kakkar

Conteúdo

1) O que é API de visualização.....	2
O que é necessário para fazer nosso mashup funcionar	2
2) Quais arquivos temos no mashup qlik básico	2
3) O que o arquivo CSS inclui	2
4) O que o arquivo HTML inclui	2
5) O que o arquivo JS inclui	2
6) O que o arquivo QEXT inclui	2
7) O que o arquivo WBL inclui	3
8) Extensão e Mashup	4
9) Conexão Mashup com Dados do Qlik Sense	4
O que saber antes de fazer um Qlik Sense Mashup	5
APIs usadas para construir mashups.	5
10) Estruturas CSS	6
Atualizando links CSS em mashups existentes	6
11) Como fazer um Mashup básico no qlik sense.....	7
Visão geral do editor de mashup	8
12) Como fazer uma extensão básica no qlik sense.....	9
Definição de propriedade em extensão.....	10
Exemplo de saída de log	11
APIs usadas para construir extensão	12
Mais links para saber sobre extensões.....	12
13) Soluções de autenticação	12
14) Licença e acesso ao mashup.....	12
O que significa On Fly	13
15) Dicas e truques	13
Como identificar qual extensão você está usando no painel do Qlik Sense usando o navegador Chrome	13
Melhore a apresentação do seu código	13
16) Métodos importantes ou métodos usados com frequência.....	14

O que é API de visualização

A interface de visualização é a interface externa para visualizações do Qlik Sense. Ela permite que os desenvolvedores obtenham visualizações definidas em um aplicativo e criem **visualizações temporárias** na hora.

A API de visualização consiste em MASHUP, extensão e widget.

O que é necessário para fazer nosso mashup funcionar

1. Aplicativo de publicação Qlik Sense
2. Um **modelo base** final ou design **de IU** para seu mashup
3. Conhecimento básico de HTML e CSS
4. Conhecimento de Bootstrap / CSS personalizado

Quais arquivos temos no mashup qlik básico

5. Arquivo CSS
6. Arquivo HTML
7. Arquivo JS
8. Arquivo QEXT
9. Arquivo WBL

O que o arquivo CSS inclui

Cascading Style Sheets (CSS) é uma linguagem de folha de estilo usada para descrever a apresentação de um documento escrito em uma linguagem de marcação, ou seja, tem um estilo definido para cada tag no arquivo HTML com base em nossa interface de usuário ou design de modelo.

O que o arquivo HTML inclui

A Linguagem de Marcação de Hipertexto (**HTML**) é a linguagem de marcação padrão para criar páginas da web e aplicativos da web com Cascading Style Sheets (CSS) e JavaScript. Ele tem todas as tags básicas baseadas em nossa UI ou Template Design.

O que o arquivo JS inclui

Arquivo **JavaScript** que contém o código **JavaScript** usado para validar o **formulário** e para definir alguma função em uma estrutura específica para usar com nosso mashup qlik. Podemos adicionar nosso próprio código **JavaScript** no mesmo arquivo JS.

O que o arquivo QEXT inclui

Metadados sobre o mashup/extensão como

10. Tipo: mashup/visualização

11. Nome: nome do mashup/visualização (pode ser alterado)
12. Versão: número da versão (pode ser alterado)
13. Descrição: do mashup/visualização (pode ser alterado) 14. autor:
autor do mashup/visualização
15. palavras-chave: qlik-sense, mashup/visualização
16. licença: -
17. repositório: link do repositório git
18. dependências: qlik-sense": ">=2.1.x"

e muitos mais.

O que o arquivo WBL inclui

Este arquivo contém todos os arquivos incluídos no mashup/extensão.

Exemplo :

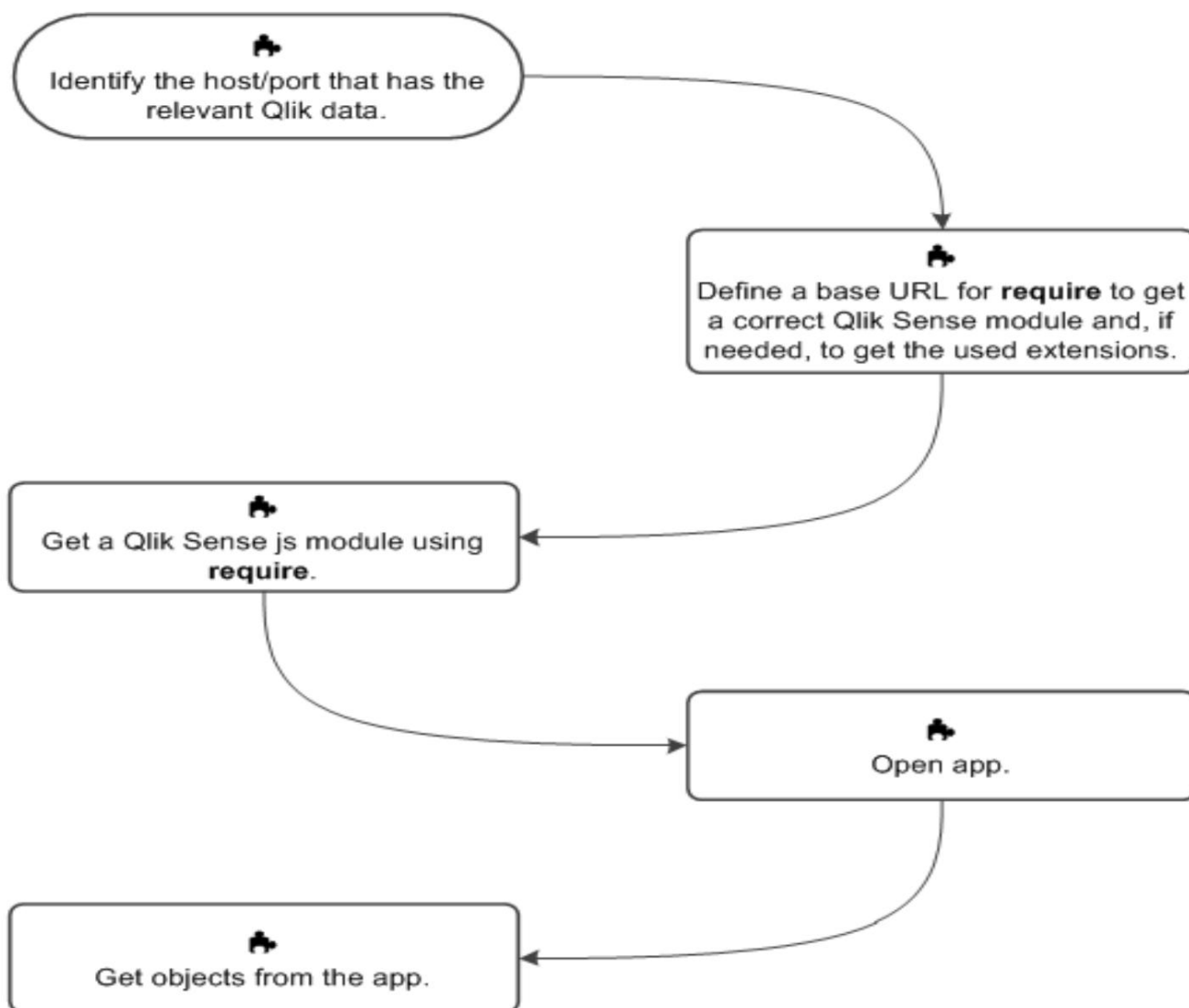
```
teste.js;  
teste.qext;  
teste.html;  
teste.css
```

Observação: os arquivos WBL e QEXT são importantes no Qlik Sense Mashup/Extension.

Extensão e Mashup

EXTENSÃO	MISTURE
É usado para adicionar mais funcionalidades/módulos em para qlik sense, pode ser usado em páginas mashup como objeto	Usa todos os objetos da planilha do aplicativo no mashup modelo
Consiste em arquivo wbl, qext, js (se angular js for usado, ele terá o arquivo template.html nele)	Consiste em arquivos wbl, qext, js, css e html
Pode tornar seu aplicativo lento dependendo da complexidade código escrito em arquivo js	Não é afetado por nenhum dos scripts, pois é web com base em que leva alguns segundos para renderizar/mostrar gráficos do aplicativo qlik sense

Conexão Mashup com Dados Qlik Sense



O que saber antes de fazer um Qlik Sense Mashup

19. Todos os arquivos base, como **PORT**, **URL**, **REQUIRE.js**, estão lá por padrão na tag **head** quando você começa a fazer um

misture

20. A função **Open App** nada mais é do que o ID do APP (somente na versão do servidor) / Nome do aplicativo (somente na versão desktop)

com caminho completo do arquivo QVF)

21. O **objeto Qlik Sense** é normalmente uma visualização de um aplicativo,

22. **Os objetos de visualização** podem ser gráficos/tabelas/mapas/extensões etc. 23. **O configurador único** é uma ferramenta do Qlik Sense que fornece uma maneira fácil de criar páginas de mashup simples

sem escrever nenhum código. Ele retorna um **objeto Qlik Sense**, que é identificado em uma URL.

24. O **configurador único** cria uma **URL** que retorna uma **página HTML completa** que contém todos os arquivos incorporados

visualização. Esta URL pode ser incorporada em uma página da web incluindo-a em um iframe.

Observação: a técnica **de configurador único** não é tão personalizável.

Exemplo :

```
<iframe src="https://sense-demo.qlik.com/single/?appid=133dab5d-8f56-4d40-b3e0a6b401391bde&obj=hRZaKk&select=Ano,2012"
frameborder="0"></iframe>
```

[Clique aqui para abrir o exemplo do QLIK](#)

APIs usadas para criar mashups.

Nome da API	Exemplo	Tipo de API
API do motor	Isso pode lhe dar todas as folhas nome, nome do aplicativo, objetos em folha etc	JSON RPC (Procedimento Remoto Protocolo de chamada codificado em JSON)
API de back-end	Todas as atividades de backend	Biblioteca JavaScript
API raiz		Biblioteca JavaScript
API do aplicativo	Operações em nível de aplicativo	Biblioteca JavaScript
API de favoritos	Manipulação de marcadores criar, apagar, ativar	Biblioteca JavaScript
API de campo	Todas as atividades de campo	Biblioteca JavaScript
API global		Biblioteca JavaScript
API de tabela	Pode fazer uma tabela personalizada	Biblioteca JavaScript
API de seleção	Pode fazer barra de seleção, pode usar funções para fazer visualizações	Biblioteca JavaScript
API variável	Criar, ler variáveis	Biblioteca JavaScript
API de visualização	Crie sua própria visualização	Biblioteca JavaScript
API de integração de aplicativos	Integra o aplicativo em mashup ou extensões	Integração de URL

API de integração única	Configurador único	Integração de URL
qlik-visual	Uma maneira de fazer visualização na mosca	Componente da Web

Estruturas CSS

As seguintes estruturas CSS funcionam com os Qlik Sense Mashups, o que significa que não deve haver conflitos entre os estilos CSS do Qlik Sense e os estilos incluídos na estrutura CSS comum:

- Bootstrap (<http://getbootstrap.com/>)
- Fundação (<http://foundation.zurb.com/>)
- Interface de usuário jQuery (<https://jqueryui.com/>)
- Design de materiais (<https://design.google.com/>)
- Interface de usuário semântica (<http://semantic-ui.com/>)
- CSS puro (<http://purecss.io/>)

Adicione a seguinte referência aos seus mashups:

```
<link rel="folha de estilo" href="../../recursos/autogerado/qlik-styles.css">
```

Atualizando links CSS em mashups existentes

Se seus mashups foram criados usando o Qlik Sense 2.1 ou anterior, você deve substituir as seguintes referências:

```
<link rel="folha de estilo" href="../../recursos/autogerado/qlikui.css">
```

```
<link rel="folha de estilo" href="../../recursos/ativos/cliente/cliente.css" media="todos">
```

com o seguinte:

```
<link rel="folha de estilo" href="../../recursos/autogerado/qlik-styles.css">
```

Observação :

- Essas estruturas de API/CSS são usadas durante a construção do seu mashup e podem ser usadas para mais personalizações na sua página.
- Quase tudo sai da caixa quando você começa a criar um novo mashup.
- Temos um exemplo de CSS do Bootstrap para começar.

Como fazer um Mashup básico no qlik sense

25. Aplicativo de publicação Qlik Sense

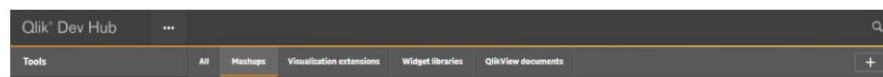
26. Um **modelo base** final ou design **de IU** para seu mashup

27. Conhecimento básico de HTML e CSS

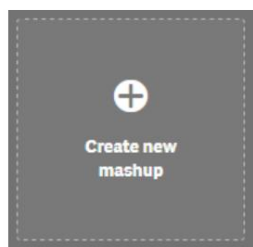
28. Conhecimento de Bootstrap / CSS personalizado

29. Acesse <http://localhost:4848/dev-hub/>

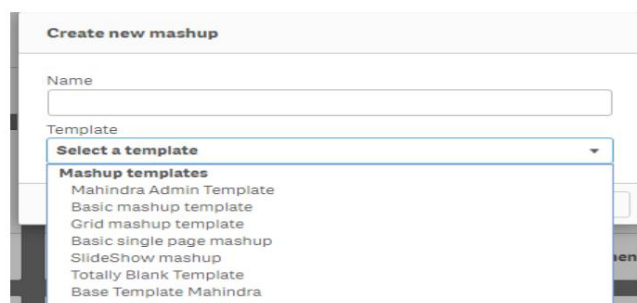
30. Em seguida, navegue para **criar um novo botão/sinal de mais** no Conner do lado direito



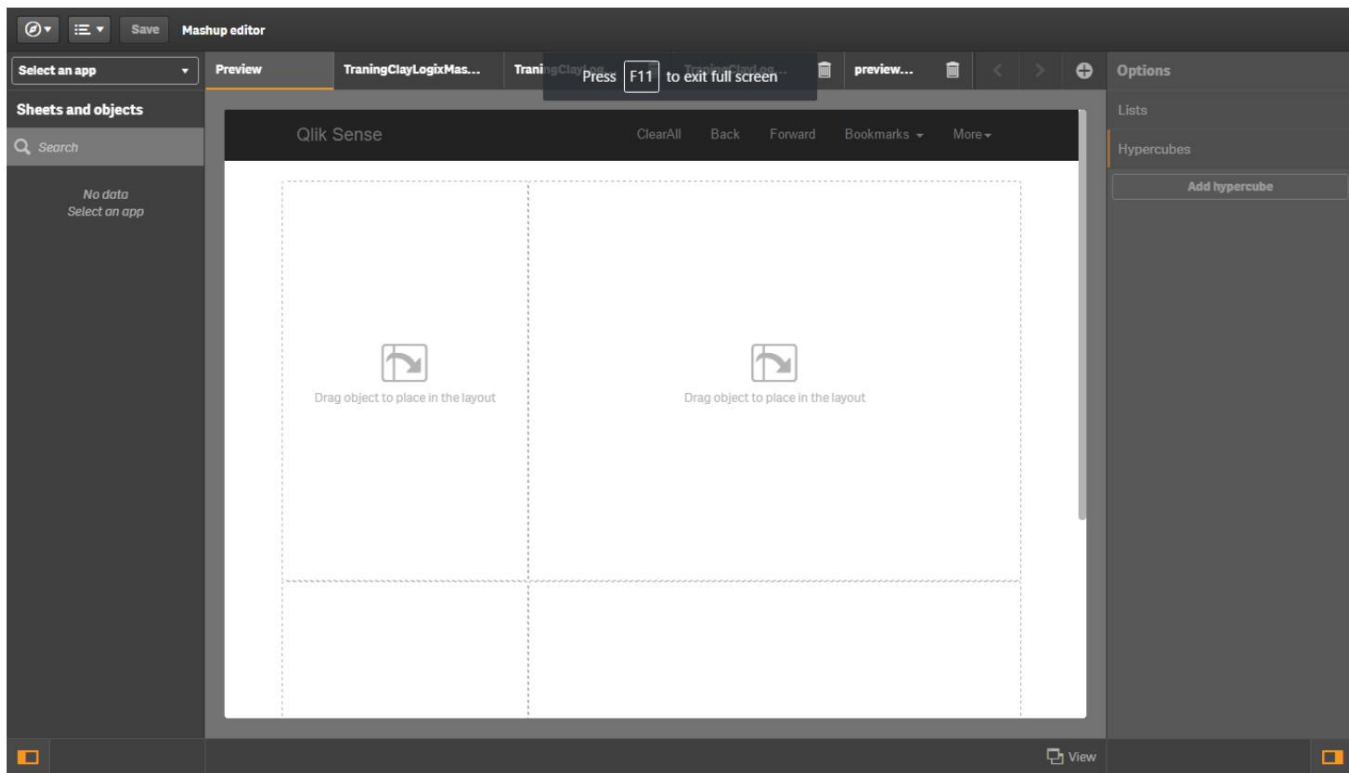
31. De qualquer forma, clique no botão criar novo mashup



32. Digite o nome preferido do mashup e selecione o tipo na opção fornecida



33. Vou escolher Grid Mashup e nomeá-lo como TraningMashup



Visão geral do editor de mashup

34. Selecione o aplicativo publicado/desenvolvido
 35. Você verá todas as folhas e objetos dentro do painel esquerdo
 36. A pesquisa pode ser usada para encontrar o objeto específico (o título é usado para pesquisar, ou seja, adicionar um título é recomendado para que ele é mais fácil identificar o objeto)
 37. Todos os gráficos/tabelas/extensões ou qualquer tipo de visualização podem ser adicionados ao mashup simplesmente arrastando e colocando-o em **qvplaceholder**
 38. **qvplaceholder** se torna **qvobject** quando um **objeto** é adicionado a ele 39.
- qvplaceholder e qvobject são uma parte importante do mashup que nos permite identificar onde queremos arrastar e largar nosso objeto
40. **qvplaceholder** é onde arrastamos nosso objeto
 41. **qvobject** é onde nosso objeto é finalmente colocado
 42. No arquivo html existem tags de divisão <div> com alguns atributos
 - Classe (**qvplaceholder/qvobject** pode adicionar outras classes personalizadas se desejar)
 - Id (isso é importante para adicionar o objeto, pois funciona como uma identidade única, dois mesmos id em um único mashup não pode ser usado, se usado um deles ficará em branco)
 - Estilo (opcional)
 43. Este processo acima preenche automaticamente o arquivo **JavaScript** para nós
 44. Pode ser feito manualmente, mas não é recomendado para alunos
 45. Você pode usar muitos IDEs como **Notepad++**, **NetBeans**, **Eclipse** etc, para **editar** todos esses arquivos como esta edição a tela ainda não foi desenvolvida ou não é amigável para o desenvolvimento da codificação

46. Vá para Melhorar a Apresentação vá para a **Página 10 (Dicas e Truques)**

Como fazer uma extensão básica no qlik sense

47. Aplicativo de publicação Qlik Sense

48. Um **modelo base** final ou design **de IU** para seu mashup

49. Conhecimento básico de HTML e CSS

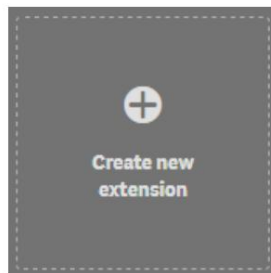
50. Conhecimento de Bootstrap / CSS personalizado

51. Acesse <http://localhost:4848/dev-hub/>

52. Em seguida, navegue para **criar um novo botão/sinal de mais** no Conner do lado direito

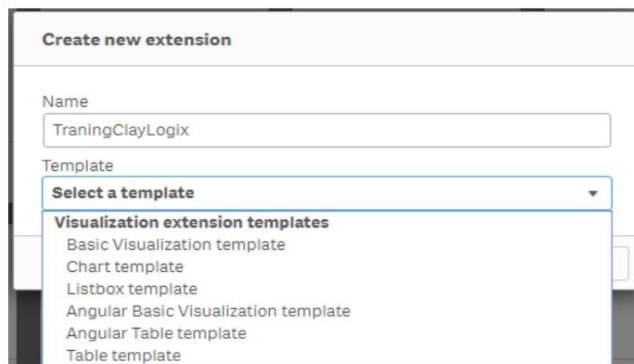


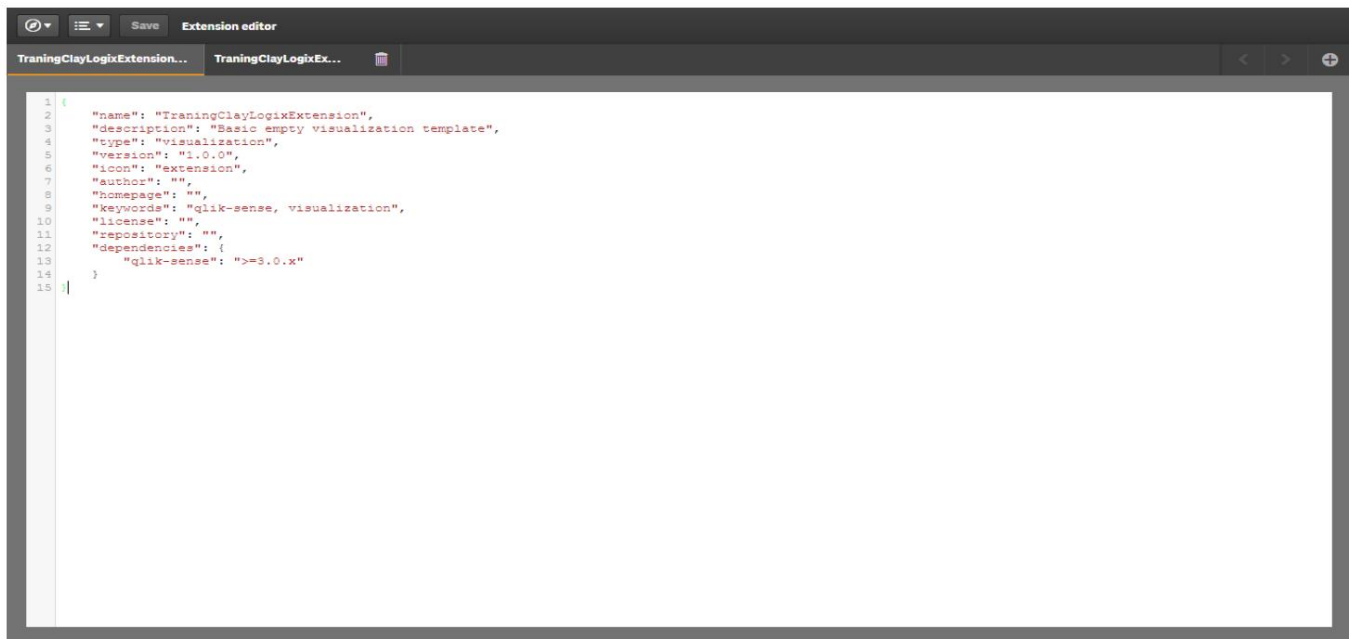
53. De qualquer forma, clique no botão criar nova extensão



54. Digite o nome preferido da extensão e selecione o tipo na opção fornecida

55. Escolherei o modelo de visualização básico e o nomearei como TraningExtension





Definição de propriedade em extensão

- **nome** = TraningExtension,
- **descrição** = Modelo básico de visualização vazia,
- **tipo** = visualização,
- **versão** = 1.0.0,
- **ícone** = extensão,
- **pré-visualização** = Logo.png
- **autor** = Ajay.R.Kakkar,
- **palavras-chave** = qlik-sense, visualização,
- **licença** = 1.0,
- **repositório** = link do repositório aqui,
- **dependências**: { qlik-sense=>=3.0.x }

Observação :

qlik-sense=>=3.0.x significa que a versão atual é igual ou superior a 3.0

a visualização personalizada deve ter 140 px por 140 px de tamanho.

Saída de log de amostra

```
extensãoMeta:Objeto
autor:""
dependências:Objeto
descrição:"Modelo de visualização básica vazia" página
inicial:""
icone: "extensão"
isLibraryItem:verdadeir
éTerceiroParte:verdadeir
palavras-chave: "qlik-sense,visualização"
licença:""
nome: "TraningExtension"
repositório:""
modelo: "TraningExtension"
templatelconClassName:"extensão-icone"
tipo:"visualização"
versão:"1.0.0"
__proto__:Objeto
nota de rodapé:""
permissões:(...)
qlInfo:Objeto
qlId:"CzXPEw"
qType:" Extensão de treinamento "
__proto__:Objeto
qSelectionInfo:Objeto
mostrarDetalhes:falso
mostrarTitulos:falso
legenda:""
titulo:""
versão:"1.0.0"
visualização:"TraningExtension"
obter permissões:()
__proto__:Objeto
```

Nota: Para ver mais detalhes, adicione a extensão em **Documents\Qlik\Sense\Extensions** e abra no dev-hub

APIs usadas para construir extensão

Nome da API	Tipo de API
API do motor	JSON RPC (protocolo de chamada de procedimento remoto codificado em JSON)
API de back-end	Biblioteca JavaScript
API raiz	Biblioteca JavaScript
API do aplicativo	Biblioteca JavaScript
API de favoritos	Biblioteca JavaScript
API de campo	Biblioteca JavaScript
API global	Biblioteca JavaScript
API de tabela	Biblioteca JavaScript
API de navegação	Biblioteca JavaScript
API de seleção	Biblioteca JavaScript
API variável	Biblioteca JavaScript

Mais links para saber sobre extensões

- [Começando](#)
- [Ferramentas de depuração de extensões e desenvolvedor web](#)
- [Introdução de componentes personalizados](#)
- [Exemplos](#)

Soluções de autenticação

1. [Solução de tickets](#)
2. [Solução de sessão](#)
3. [Solução de cabeçalho](#)
4. [SAML](#)
5. [Usuários anônimos](#)

Licença e acesso ao mashup

1. Não há licença se você estiver desenvolvendo no mesmo host ou domínio
2. A licença se aplica quando você sai do host ou domínio (para uso público ou para o mundo acessá-lo), se 1000 pessoas estão atingindo ou usando seu servidor Qlik e usando toda a memória, seu mashup Qlik tende a ficar lento

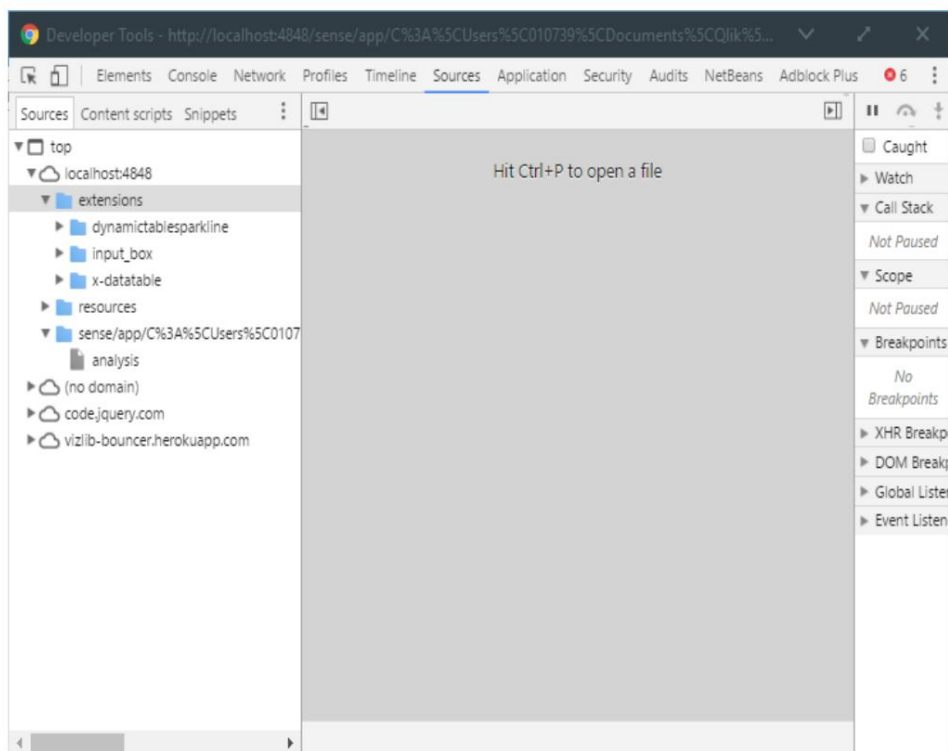
O que significa On Fly

Isso significa que podemos desenvolver mapas ou gráficos diretamente em páginas de mashup que são renderizadas **temporariamente** e destruídas quando página ou seção está fechada. Este método leva tempo para construir ou codificar nossos gráficos / visualização em comparação com o qlik sense eixo.

Dicas e truques

Como identificar qual extensão você está usando no painel do Qlik Sense usando o Chrome navegador

1. abra a planilha na qual você deseja identificar a extensão
2. assim que a planilha for carregada, pressione **F12** e você verá uma aba onde haverá muitas abas de opções
3. selecione **FONTES** e você obterá todos os arquivos na URL carregados como visualização em árvore/explorer
4. Haverá uma pasta chamada extensão na qual você obterá todos **os nomes de extração e os arquivos renderizados em formato de pasta**



Melhore a apresentação do seu código

5. Para embelezar seu arquivo JavaScript
<http://jsbeautifier.org/>

6. Para embelezar seu arquivo CSS

<http://www.cleancss.com/css-beautify/>

7. Para embelezar seu arquivo HTML

<http://www.cleancss.com/html-beautify/>

Métodos importantes ou métodos usados com frequência

1. **qlik.currApp()** : Referência ao aplicativo atual.
2. **qlik.getAppList()** : Obtém uma lista de aplicativos Qlik Sense aos quais você tem acesso.
3. **qlik.getExtensionList()** : Obtenha uma lista de extensões no seu servidor qlik sense.
4. **qlik.resize()** : Atualiza todos os objetos ou um objeto específico adicionando o ID do objeto como parâmetro.
5. **qlik.app.back()** : Seleciona a última seleção ou a seleção anterior.
6. **app.clearAll()** : Limpa toda a seleção.
7. **app.forward()** : Seleciona a seleção de avanço.
8. **app.field('[Nome do campo]')** : Obtém uma referência de campo e adiciona o uso de métodos da API de campo.
9. **app.field('[Nome do campo]').clear()** : Obtém uma referência de campo e limpa a seleção de campo.
10. **app.lockAll()** : Bloqueia todas as seleções no aplicativo.
11. **app.unlockAll()** : Desbloqueia todas as seleções no aplicativo.
12. **app.field('[Nome do campo]').clearOther(true)** : Limpa outros campos e adiciona o parâmetro true se os campos bloqueados devem ser limpo
13. **app.field('[Nome do campo]').select()** : Seleciona valores de campo, parâmetros select(Array:[selecionar índice de valor número], alternar: [verdadeiro/falso], softlock: [verdadeiro/falso])
14. **qlik.app.field('[Nome do campo]').selectAll()** : Selecione todos os valores no campo, para softlock adicione o parâmetro true.
15. **qlik.app.field('[Nome do campo]').selectAlternative()** : Seleciona valores alternativos em um campo, para softlock add parâmetro verdadeiro.
16. **qlik.app.field('[Nome do campo]').selectExcluded()** : Seleciona valores excluídos em um campo, para softlock add parâmetro verdadeiro.
17. **qlik.app.field('[Nome do campo]').selectPossible()** : Seleciona valores possíveis em um campo, para softlock add parâmetro verdadeiro.
18. **qlik.app.field('[Nome do campo]').toggleSelect(Valor, softlock:[true/false])** : Alternar seleção de campo, adicionar Valor como string.
19. **qlik.app.field('[Nome do campo]').selectMatch(Value, softlock:[true/false])** : Formulário Selecionar valores correspondentes o campo.
20. **qlik.navigation.getCurrentSheetId()** : Obter ID da planilha atual.
21. **qlik.navigation.getMode()** : Obter modo de edição/análise de planilha.
22. **qlik.navigation.setMode()** : Defina o modo de planilha para Editar [qlik.navigation.EDIT] /Analysis [qlik.navigation.ANÁLISE].
23. **qlik.navigation.gotoSheet(Value)**: Navegue até a planilha e substitua Value por SheetID.
24. **qlik.navigation.gotoStory(Value)**: Navegue até a história e substitua Value por StoryID.

- 25. **qlik.navigation.isModeAllowed(mode:[edit/analysis])** : verifica se um determinado modo é permitido.
- 26. **qlik.navigation.nextSheet()** : vai para a próxima planilha.
- 27. **qlik.navigation.prevSheet()** : vai para a planilha anterior.
- 28. **app.variable.getByNome('var_name')** : obtém metadados da variável.
- 29. **qlik.app.variable.getContent('var_name', callbackfunction)** : obtém conteúdo da variável.
- 30. **qlik.app.variable.setContent('var_name', 'value')** : define o conteúdo da variável.
- 31. **qlik.app.variable.setNumValue('var_name', 100)** : define o valor numérico da variável.
- 32. **qlik.app.variable.setStringValue('var_name', 'value')** : define o valor da string da variável.
- 33. **exportData()** : este método ajuda na exportação de dados. A exportação padrão é no Excel.