

Bônus: Instalando a app com Web App Manifest

Transcrição

Mencionei inúmeras vezes o fato de que nosso site é uma app, mas ele continua sendo um site, precisamos digitar um endereço na barra do navegador, não sendo passível de instalação em nosso computador ou celular.

Já realizamos instalação da app, armazenando seus arquivos no Cache Storage, mas quando falo em instalar, agora, significa conseguir acessar a aplicação sem que ele passe pelo navegador. Quero ter um ícone na área de trabalho, a partir do qual consigo acessá-la, sem a necessidade de fazer isto como se estivesse acessando um site.

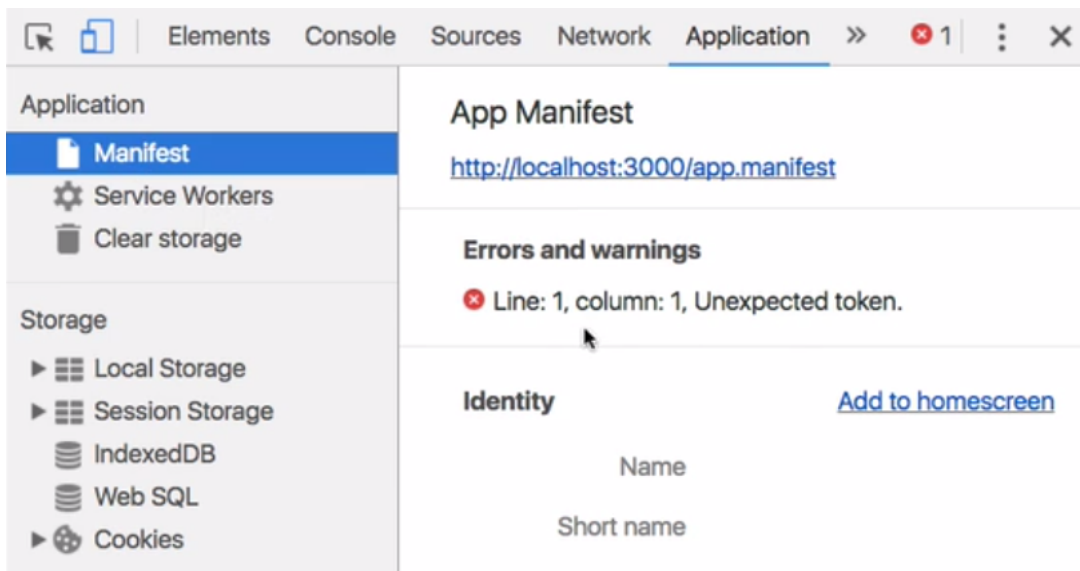
Se você quiser transformar "ceep" em uma app depois de tudo o que fizemos, precisamos escrever para a aplicação informações como o ícone a ser utilizado, nome (uma vez que não existe mais url a ser acessada) e outras especificações.

Estas informações precisam ser repassadas ao "Manifest", um arquivo que vai descrever a aplicação. Para tal, criaremos um arquivo novo intitulado `app.manifest`, sendo que o nome não é tão relevante. O importante é alterar o `index.html` com um link que aponta para um `manifest`:

```
<!DOCTYPE html>
<html lang="pt-br" manifest="offline.manifest">
<head>
  <link rel="manifest" href="app.manifest">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="css/opcoesDaPagina.css">
  <link rel="stylesheet" href="css/opcoesDoCartao.css">
  <link rel="stylesheet" href="css/cabecalho.css">
  <link rel="stylesheet" href="css/login.css">
  <link rel="stylesheet" href="css/loginForm.css">
  <link rel="stylesheet" href="css/loginStatus.css">
  <link rel="stylesheet" href="css/cartao.css">
  <link rel="stylesheet" href="css/novoCartao.css">
  <link rel="stylesheet" href="css/mural.css">
  <title>Ceep</title>
</head>
```

Feito isto, recarregarei a página e, caso queira visualizar alguma alteração, precisaremos modificar o Service Worker, portanto abriremos o arquivo `service-worker.js` para atualizar sua versão, fazendo o mesmo em `offline.manifest`, como já fizemos em vídeos anteriores.

Agora, conseguiremos ver o Service Worker novo sendo instalado, o ativarei, e teremos um "Manifest", que nos mostra um erro:



O arquivo `manifest` não está em branco e, neste caso, trata-se de um JSON, arquivo objeto de JAVASCRIPT. Vamos reabrir o arquivo e acrescentar estas linhas:

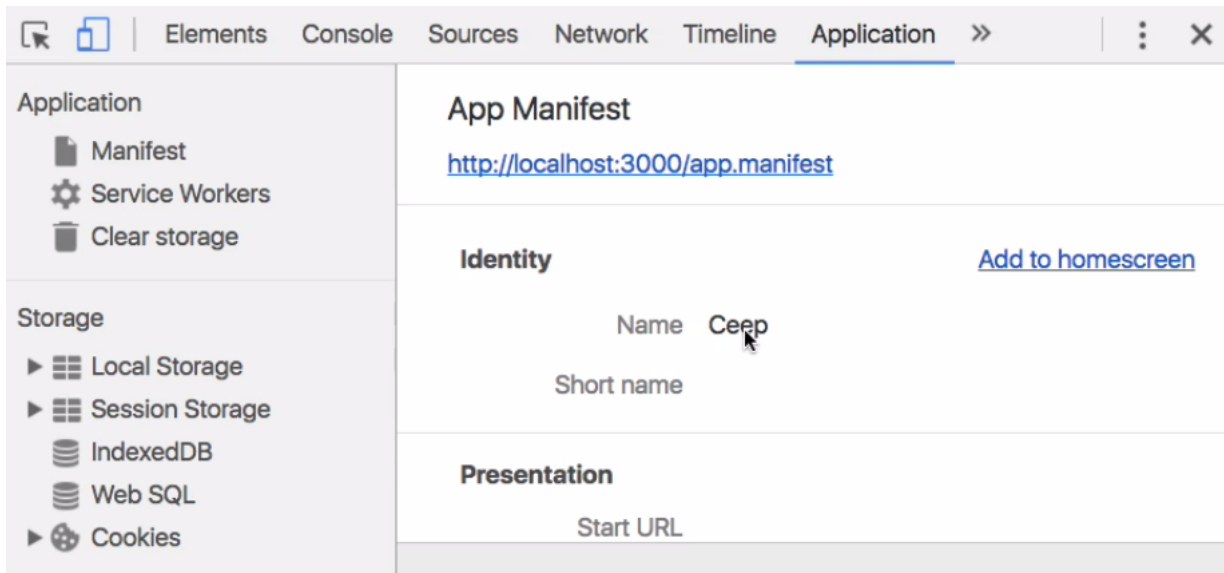
```
{  
  
}
```

Já temos um `manifest` vazio, portanto recarregarei a página, cliquei em "Manifest". Não houve erro nenhum, ou seja, ele está válido. Será que já consigo instalar o aplicativo no meu computador? Podemos tentar, indo à aba "Application", na parte de "Manifest" e clicando no link "*Add to homescreen*". O que isto vai fazer é adicionar nossa app na página do usuário, na *home* (página inicial).

O Console nos mostra outro erro, e seguiremos seus passos até conseguirmos instalar a aplicação. Ele nos diz que o site não pôde ser instalado pois o `manifest` não foi carregado por estar vazio, não ter nenhuma informação dentro dele. Vamos dar um nome à aplicação:

```
{  
  "name": "Ceep"  
}
```

Recarregaremos a página e obtivemos a informação de que seu nome existe:



Clico em "Add to homescreen" e somos informados sobre mais um erro ("Site cannot be installed: the start URL in manifest is not valid", ou seja, "A app não pôde ser instalada pois a *start url* não é válida"). Esta *start url* é o local onde os arquivos serão procurados inicialmente.

Por exemplo, a barra está procurando a partir de onde? Qual a pasta raiz? Como o `manifest` se encontra na pasta raiz do projeto, direi que a `"start_url"` dele é a pasta em que estou no momento:

```
{
  "name": "Ceep",
  "start_url": "./"
}
```

Este já é um `manifest` válido. Como tenho duas propriedades agora, separo-as por vírgulas. Recarregaremos a página, daremos uma olhada na aba "Application", a qual especifica que a *Start URL* é, de fato, `"./"`. O `manifest` está sendo alterado. Abriremos o Console e verificaremos o que ocorre quando clicar em "Add to homescreen". Mais uma vez, um erro: "Site cannot be installed: the manifest display property must be set to 'standalone' or 'fullscreen", ou seja, a propriedade `display` não foi colocada.

Ao trabalharmos em uma aplicação, temos que explicitar que queremos que ela seja exibida fora do navegador, ou que ela apareça como uma aba do navegador. Para isto, precisaremos acrescentar mais informações ao arquivo `app.manifest`:

```
{
  "name": "Ceep",
  "start_url": "./",
  "display": "standalone"
}
```

Em que `"standalone"` informa que queremos que a app rode como se fosse nativa, sem necessidade do uso de abas do navegador. Carregaremos a página novamente, tentarei adicionar à página principal ("Add to homescreen"). Ainda encontra-se um erro: não temos um ícone a ser utilizado. Seu tamanho deve ter pelo menos `144px` de largura, e ele deve ser em extensão `.png`.

Já tenho um ícone pronto para uso, mas quem quiser pode utilizar qualquer um que preferir. Precisamos inseri-lo no projeto, na pasta "img" [deste arquivo zipado \(https://github.com/alura-cursos/ceep-pwa/archive/v6.1.zip\)](https://github.com/alura-cursos/ceep-pwa/archive/v6.1.zip) disponibilizado aqui no

curso. O código, portanto, ficará assim:

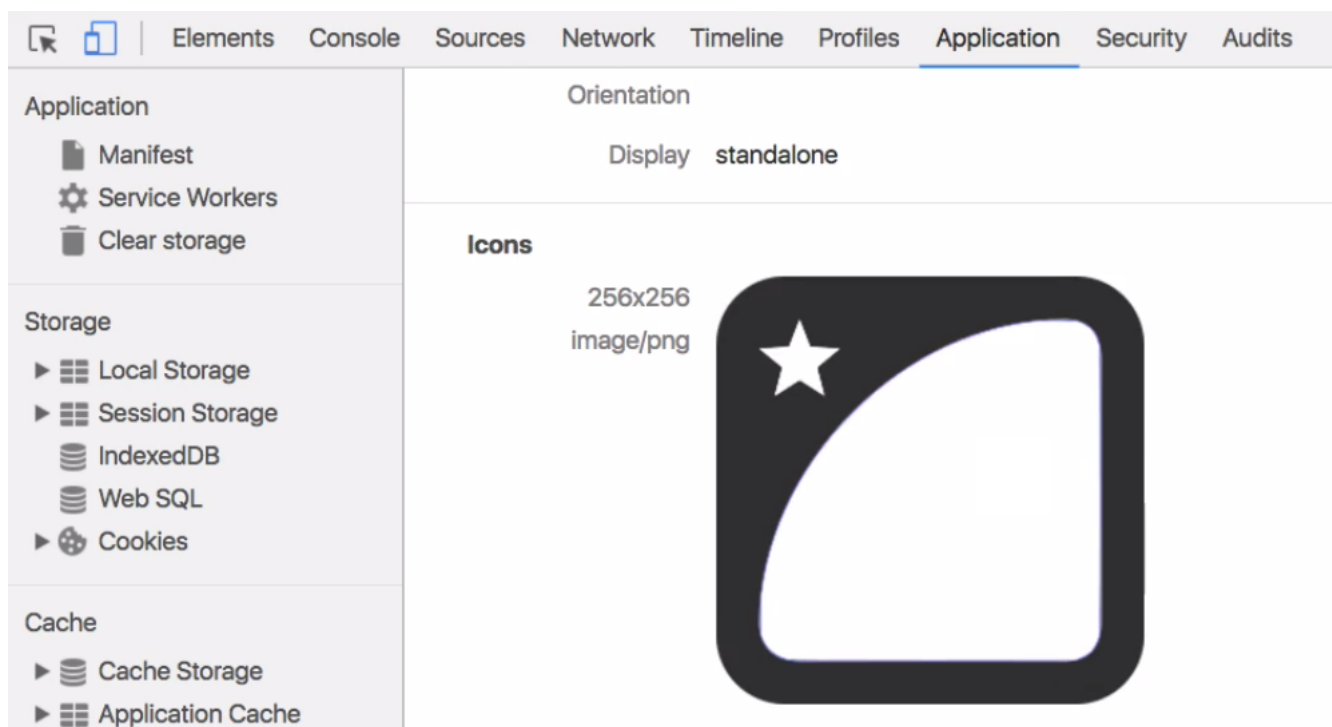
```
{
  "name": "Ceep",
  "start_url": "./",
  "display": "standalone",
  "icons": [

  ]
}
```

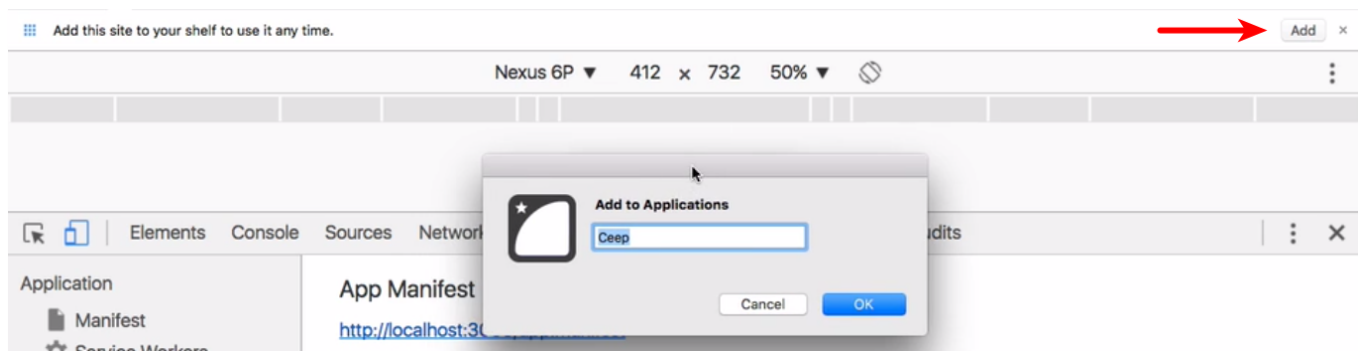
Poderemos acrescentar quantos ícones quisermos, e o valor colocado no código é uma lista com vários tamanhos possíveis deles. Pode-se utilizar ícones de 32px a 256px. Como possuímos apenas um ícone, colocaremos as várias informações referentes a ela: seu tamanho, url (a `src` da imagem) e tipo (o formato dela é importante!).

```
{
  "name": "Ceep",
  "start_url": "./",
  "display": "standalone",
  "icons": [
    {
      "src": "img/caelum-logo-256.png",
      "sizes": "256x256",
      "type": "img/png"
    }
  ]
}
```

São estas as informações de que precisamos para cada ícone que quisermos ter na página, cujos tamanho e tipo podem variar. Vamos recarregar a página, e o ícone é identificado.



Caso tentemos adicionar à *homescreen*, aparece um aviso no topo da página: *"Add this site to your shelf to use it any time."*, uma forma de dizer que a app foi adicionada com sucesso. Cliquei em "Add" para confirmar esta ação, existindo inclusive a opção de escolher o nome que quero usar.



Agora temos um ícone na pasta como se tivesse um aplicativo instalado e nativo. Vou passá-lo ao Desktop, tendo acesso efetivo à aplicação clicando no ícone. Criar um arquivo de `manifest` é interessante pois, com poucas informações, conseguimos ter uma experiência praticamente igual à de um aplicativo nativo, algo que não ocorre apenas no Desktop, funcionando também em um smartphone, a partir de sua instalação.

Não é o usuário que terá que acionar o *"Add to homescreen"* já que, automaticamente, o navegador vai saber quando o mesmo site está sendo acessado com frequência. Aquela mensagem que pergunta a ele se quer adicionar a aplicação será mostrada ao usuário, ficando a seu critério decidir se quer fazê-lo ou não. A versão Desktop ainda está sendo elaborada, porém a do smartphone funciona perfeitamente.