Disciplina: Aplicações Ricas

Diretoria Acadêmica
Centro Universitário Senac

Prof. **Mario L. P. Toledo**

Aula 7 - Geolocation

Obtendo a Localização

- O que é geolocalização?
  - Permite obter a localização do usuário
- Muitos aplicativos utilizam geolocalização
  - Exemplos?
- Alguns sites também passaram a utilizar a localização do usuário
  - Qual a utilidade?

Obtendo a Localização

- A API de geolocalização pertence ao navegador
  - É padronizada
- Objeto geolocation do navigator
- Os métodos principais do geolocation são:
  - getCurrentPosition()
  - watchPosition()

Obtendo a Localização

- Nem todo navegador implementa geolocation. É importante checar se ele existe:

```
if ("geolocation" in navigator) {
  /* geolocation is available */
} else {
  alert("Ops, seu navegador não suporta geolocation");
}
```

Obtendo a Localização

- Para obter a localização, utilizar o método getCurrentPosition()
- Aceita 3 parâmetros:
  - O callback em caso de sucesso
  - O callback em caso de erro
  - Opções de geolocalização
- O que é callback?
  - É uma função que será chamada ao final da execução
  - Deve ser implementada

Obtendo a Localização

- O callback de sucesso recebe um parâmetro position
  - Possui o objeto coords, que contém os valores latitude, longitude e accuracy
- O callback de erro recebe um parâmetro error, o qual contém o atributo code, que é o código de erro
  - O código de erro pode ser:
    - 1: permissão negada
    - 2: posição indisponível
    - 3: timeout

Obtendo a Localização

- As opções de geolocalização são:
  - enableHighAccuracy: habilitar alta precisão (padrão: false)
  - timeout: tempo em milissegundos (padrão: infinito)
  - maximumAge: por quanto tempo o navegador pode usar uma localização obtida e armazenada em cache, em milissegundos (padrão: 0)
- É necessário criar um objeto com os parâmetros que se quer configurar

Obtendo a Localização

- Nenhum dos parâmetros é obrigatório
  - Mas sem o callback de sucesso, a chamada é inútil

Obtendo a Localização

- Como fazer uma função JavaScript para pegar a localização?

```
navigator.geolocation.getCurrentPosition(function(position) {
        //fazer algo
});
```

Obtendo a Localização

- Como informar o usuário em caso de erro?

```
navigator.geolocation.getCurrentPosition(function(position) {
        //fazer algo
}, function(error){
        alert(“Ops, ocorreu um erro ao obter sua localizacao”)
});
```

Obtendo a Localização

- Como configurar um timeout?

```
navigator.geolocation.getCurrentPosition(function(position) {
        //fazer algo
}, function(error){
        alert("Ops, ocorreu um erro ao obter sua localizacao")
}, {enableHighAccuracy:true, maximumAge:30000, timeout:27000});
```

Localização Contínua

- O que é localização contínua?
  - Fica constantemente solicitando a localização
  - Permite rastrear o usuário
- Por que isso seria útil?

Localização Contínua

- Utiliza-se o método watchPosition()
  - Possui os mesmos parâmetros que o getCurrentPosition()
  - O callback de sucesso é chamado sempre que há uma mudança na localização

Distância Entre Posições

- A API LatLon permite calcular distância entre duas coordenadas:
  - http://www.movable-type.co.uk/scripts/latlong.html
- Basta importar o script na página e criar um objeto do tipo LatLon passando a latitude e longitude:
  - var posicao1 = new LatLon(position.coords.latitude, position.coords.longitude);
- Chamar a função distanceTo() passando um segundo objeto LatLon com outra coordenada:
  - var distancia = posicao1.distanceTo(posicao2);

Distância Entre Posições

```
function getDistanceBetweenTwoPoints(cord1, cord2) {
  if (cord1.lat == cord2.lat && cord1.lon == cord2.lon) {
    return 0;
  }

  const radlat1 = (Math.PI * cord1.lat) / 180;
  const radlat2 = (Math.PI * cord2.lat) / 180;

  const theta = cord1.lon - cord2.lon;
  const radtheta = (Math.PI * theta) / 180;

  let dist =
    Math.sin(radlat1) * Math.sin(radlat2) +
    Math.cos(radlat1) * Math.cos(radlat2) * Math.cos(radtheta);

  if (dist > 1) {
    dist = 1;
  }

  dist = Math.acos(dist);
  dist = (dist * 180) / Math.PI;
  dist = dist * 60 * 1.1515;
  dist = dist * 1.609344; //convert miles to km

  return dist;
}
```

Google Maps

- É possível exibir a posição em um mapa estático do Google Maps
  - Colocar no src da imagem o endereço:
    - http://maps.google.com/maps/api/staticmap
      ?center=latitude,longitude
      &zoom=12&size=440x440&maptype=roadmap
      &markers=color:red|color:red|label:a|latitude,longitude
      &sensor=false&key=CHAVE_API&signature=ASSINATURA

Google Maps

- A API do Google Maps permite fazer coisas muito mais legais
  - Você pode usar gratuitamente a API via JavaScript
    - https://developers.google.com/maps/web/
  - É necessário ativar o uso da API para obter uma chave
  - Importar o script na página passando a sua chave:
    - <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=sua_chave"/>

Google Maps

- Para adicionar um mapa na página:
  - Criar um novo objeto google.maps.Map passando a div onde o mapa fic as opções
  - As principais opções são o centro do mapa (latitude e longitude) e o zoo
    - var mapaDiv = document.getElementById('map');
      var mapOptions = {center: { lat: latitude, lng: longitude}, zoom: 14};
      var meuMapa = new google.maps.Map(mapaDiv, mapOptions);

Google Maps

- Para adicionar um marcador no mapa:
  - Crie um objeto google.maps.Marker passando as opções
  - As principais opções são a posição (latitude e longitude) e o objeto de mapa
    - var mOptions = {position: { lat: latitude, lng: longitude}, map: meuMapa};
      var marker = new google.maps.Marker(mOptions);

Google Maps

- Ver a API completa no site do Google:
  - https://developers.google.com/maps/documentation/javascript/

Exercícios

- Crie uma página que **continuamente atualize sua posição** em um mapa do Google Maps
  - No mapa, adicione dois marcadores: um na sua casa e outro onde você está
  - Exiba em uma div separada a distância que você está de casa