

Google Awareness API

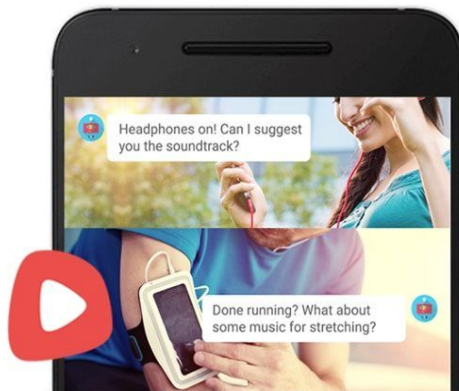
Davi Tabosa
Programação para Dispositivos Móveis

O que veremos hoje?

- O que é o Awareness API
 - Origem
 - Exemplos de utilização
 - Fences
 - Snapshots
 - Implementação
- Prática
 - Atividade no sigaa para ser entregue
 - Prints do código

API

- Google I/O 2016
- Tornar apps mais inteligentes e adaptados à sua rotina através de informações contextualizadas.



Exemplos

- Apps de meteorologia que podem lançar a previsão climática para o ChromeCast mais próximo.
- Um GPS que manda dados do trânsito para o Google Home caso o caminho que o usuário costuma seguir naquele horário esteja muito engarrafado.

Como ela funciona?

A API emprega sete parâmetros diferentes para entender o usuário:

Context type	Example
Time	Current local time
Location	Latitude and longitude
Place	Place, including place type
Activity	Detected user activity (walking, running, biking)
Beacons	Nearby beacons matching the specified namespace
Headphones	Are headphones plugged in?
Weather	Current weather conditions

Promessas

Além da principal que é a construção de apps inteligentes, eles prometem:

- “Fácil implementação”
- Melhores dados de contexto
- Bom uso de recursos do aparelho(ex: Bateria)

Fences e Snapshots

- A Awareness API consiste em duas APIs distintas que seu aplicativo pode usar para obter sinais de contexto para determinar a situação atual do usuário:

- Fences API:

Permite que seu aplicativo reaja à situação atual do usuário e fornece notificação quando uma combinação de condições de contexto é atendida.

Ex: faça isso quando o usuário conectar o fone

- Snapshots API

Permite que seu aplicativo solicite informações sobre o contexto atual do usuário.

Por exemplo, "me informe a localização atual do usuário e as condições do tempo atual".

Fences e Snapshots

- Tipos de Fences e Snapshots disponíveis por cada API
- Fences API
 - Beacons
 - Headphone
 - Location
 - Time:
 - Activity Detection
- Snapshots API
 - Beacons
 - Headphone
 - Location
 - Places
 - Weather

Permissões Snapshot API

Method	Required Android Permission
<code>getDetectedActivity()</code>	<code>com.google.android.gms.permission.ACTIVITY_RECOGNITION</code>
<code>getBeaconState()</code>	<code>android.permission.ACCESS_FINE_LOCATION</code>
<code>getHeadphoneState()</code>	none
<code>getLocation()</code>	<code>android.permission.ACCESS_FINE_LOCATION</code>
<code>getPlaces()</code>	<code>android.permission.ACCESS_FINE_LOCATION</code>
<code>getWeather()</code>	<code>android.permission.ACCESS_FINE_LOCATION</code>

Permissões Fence API

Fence API fence type	Required Android permission
DetectedActivityFence	com.google.android.gms.permission.ACTIVITY_RECOGNITION
BeaconFence	android.permission.ACCESS_FINE_LOCATION
HeadphoneFence	none
LocationFence	android.permission.ACCESS_FINE_LOCATION
TimeFence	none

Primeiros Passos

Você deve ir ao site do Awareness API para pegar uma key para acessar a API.

<https://developers.google.com/awareness/android-api/get-a-key>

Mas o projeto já está com uma key configurada ;)

GitHub

<https://github.com/davitabosa12/Awarenesstutorial>

Baixem o Projeto

Prática com Snapshot e Fences

1. Abram o Android Studio
2. Realizem os procedimentos de permissões
3. Criem uma aplicação que exibe após um clique em um botão se o headphone está ou não plugado
4. Link do Arquivo da Ativade: <https://goo.gl/j2AKJM>
5. Dicas e pescas:
<https://developers.google.com/awareness/android-api/get-started>

Passos para criar um Snapshot

Passos para usar um Snapshot:

1. Obtenha um SnapshotClient
2. Use o SnapshotClient para obter um método a sua escolha(Places,Headphones,etc)
3. Adicione um Listener para receber o resultado da sua requisição
4. Com esse listener,mande fazer algo(Toast,vibrar,acender lanterna, etc)

Observação sobre Snapshot

O Snapshot necessita de um mecanismo de
ResultCallbacks/Listener para que seja possível obter o resultado dos métodos

Exemplo Snapshot

```
private void startSnapshot() {  
    //pegar o cliente de snapshot da api  
    SnapshotClient snapshotClient = Awareness.getSnapshotClient(this);  
    snapshotClient.getWeather().addOnSuccessListener(new OnSuccessListener<WeatherResponse>() {  
        //promise da requisicao getWeather  
        @Override  
        public void onSuccess(WeatherResponse weatherResponse) {  
            //mostrar a previsao do tempo no celular  
            String resp = weatherResponse.getWeather().toString();  
            Toast.makeText(MainActivity.this, resp, Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```


Prática com Snapshot

1. Agora vamos tentar checar a localização do usuário por meio do Places
2. Dicas e pescas:
<https://developers.google.com/awareness/android-api/get-started>

Passos para criar uma Fence

Passos para usar uma Fence:

1. Criar uma Fence
2. Criar um Intent Filter
3. Registrar um Broadcast Receiver com o Intent Filter
4. Criar um Pending Intent
5. Registrar na API Awareness o Pending Intent e a Fence

Passos para criar uma Fence

```
public class FenceActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fence);

        //Criar as AwarenessFences
        AwarenessFence headphone = HeadphoneFence.during(HeadphoneState.PLUGGED_IN);

        //Filtros de Intent
        IntentFilter hp = new IntentFilter("headphone");
        //Registrar Receivers (actions) na pilha do Android
        registerReceiver(new VibrateAction(), hp);
        //Registrar PendingIntents getBroadcast com os filtros criados
        PendingIntent pi = PendingIntent.getBroadcast(this, 123, new Intent("headphone"), PendingIntent.FLAG_CANCEL_CURRENT);
        //Registro de Fences no Google Awareness API
        FenceClient fc = Awareness.getFenceClient(this);
        fc.updateFences(new FenceUpdateRequest.Builder().addFence("Headphone", headphone, pi).build());
    }
}
```

Prática com Fence

1. Modifique o código anterior de modo que ao se plugar o fone, o celular vibre
2. Crie um Fence para que o celular vibre quando o usuário estiver andando

Google Awareness API

Davi Tabosa
Programação para Dispositivos Móveis