

Criando e Adaptando Bibliotecas do Ruby

Gustavo Toshi Komura

Universidade Federal do Paraná - UFPR
Ciência da Computação
Departamento de Informática
Orientador: Professor Doutor Bruno Müller Junior

Curitiba - 11 de Dezembro de 2014

Sumário

- 1 Introdução
 - Objetivos do Trabalho
 - Funcionalidades Google-Maps-For-Rails
 - Proposta de Novas Funcionalidades
- 2 Bibliotecas do Ruby
 - Conceitos
 - Processo de Criação
 - Uso da gema `gemtranslatetoenglish`
- 3 Bibliotecas do Ruby Mapeando API Google Maps
 - API Google-Maps-For-Rails
 - Processo de Adaptação
 - API Google-Maps-For-Rails Adaptada
- 4 Conclusão

Objetivos do Trabalho

Objetivos

Apresentar:

- Conceitos sobre as bibliotecas do **Ruby**.
- O processo de criação de uma biblioteca do **Ruby**.
- O processo de adaptação de uma biblioteca do **Ruby**.
- Exemplos de uso das bibliotecas do **Ruby**.
 - Uso da gema “**gemtranslatetoenglish**”: Tradução do português para o inglês.
 - Uso da gema “**Google-Maps-For-Rails Adaptada**”: Determina direções em uma biblioteca do **Ruby** que mapeia a **API** do **Google Maps**.

Funcionalidades Google-Maps-For-Rails

Funcionalidades

Por meio de código **CoffeeScript**:

- Cria mapas do **Google Maps** em páginas web.
- Permite a criação de mais de um mapa na mesma página (id).
- Cria sobreposições para os mapas (markers, circles, polygon, polylines, etc).
- Permite a criação de várias sobreposições do mesmo tipo com a chamada de uma única função.
- Determina fronteiras no mapa, utilizando a posição de markers.
- Permite readaptação do código **core** para criar e manipular mapas de ferramentas similares ao do **Google (Bing Maps)**,

Proposta de Novas Funcionalidades

Novas Funcionalidades

Por meio de código **CoffeeScript**:

- Gerar direções entre dois lugares no mapa.
- Permitir a escolha do veículo de locomoção.
- Gerar direções entre dois lugares com pontos intermediários obrigatórios.
- Permitir a troca de cores entre os tracejados entre dois pontos do caminho.
- Permitir a escolha do tipo de caminho a ser percorrido (mais curto, mais longo, menor tempo, maior tempo, etc).

Conceitos

Formas

- Forma **gem**: Forma mais utilizada para distribuição, por isso as bibliotecas do **Ruby** são conhecidas como **gems/gemas**. Seu processo de instalação é feito por meio do programa **gem**.
- Forma **arquivo compactado** em `.tar.gz` ou `.zip`: Forma menos utilizada. Seu processo de instalação é feito por meio de arquivos `README` ou `INSTALL`.

Programa gem

- Similar ao sistema de distribuição **apt-get**.
- Executado em linhas de comando no formato "**gem <operação>**", onde "**<operação>**" pode ser:
 - **install** para fazer instalação de gemas.
 - **search** para fazer busca de gemas.

Segurança

Motivos

- 1 Abrir brechas.
- 2 Risco de perda de informações.

Processo de Segurança do RubyGems

O **RubyGems** é o repositório de distribuição padrão utilizado no programa **gem**.

- 1 Certificação da gema.
- 2 Verificação de segurança na instalação da gema.

Reportar Vulnerabilidades

- Vulnerabilidades nas gemas de outros desenvolvedores
- Vulnerabilidades das próprias gemas.

Processo de Criação

Processo

❶ Criar a estrutura automaticamente

```
1 bundle gem 'nome da gema'
```

Código 1: Cria Gema Forma Geral

- ❷ Editar “**‘nome da gema’.gemspec**” para definir as descrições da gema.
- ❸ Desenvolver códigos de funcionalidades.
- ❹ Desenvolver códigos de testes.

Uso da gema `gemtranslatetoenglish`

Adiciona gema no Gemfile

Acrescenta a gema de tradução no Gemfile do projeto

```
1 gem 'gemtranslatetoenglish'
```

Código 2: Adiciona `gemtranslatetoenglish` no Gemfile

Funcionalidade na View

Utiliza a funcionalidade de tradução da nova gema em uma view

```
1 translate("OI")
```

Código 3: Exemplo de `gemtranslatetoenglish` na View

```

1 handler = Gmaps.build('Google');
2 handler.buildMap({ provider: {}, internal: {id: 'map'}} ,
3     function(){
4         markers = handler.addMarkers([
5             {
6                 "lat": 0,
7                 "lng": 0,
8                 "picture": {
9                     "url": "https://addons.cdn.mozilla.net/img/uploads/
10                     addon_icons/13/13028-64.png",
11                     "width": 36,
12                     "height": 36
13                 },
14                 "infowindow": "hello!"
15             }
16         ]);
17         handler.bounds.extendWith(markers);
18         handler.fitMapToBounds();
19     });

```

Código 4: Exemplo CoffeeScript API Google-Maps-For-Rails

Processo de Adaptação

Processo

- 1 Realizar **engenharia reversa** para gerar diagramas de alto nível.
- 2 Entender a biblioteca por meio dos diagramas de alto nível.
- 3 Realizar adaptações.

API Google-Maps-For-Rails Adaptada

Criar Direções

Por meio de código **CoffeeScript**:

```
1 $(document).ready(->
2   handler = Gmaps.build("Google")
3   handler.buildMap
4     provider: {}
5     internal:
6       id: "map"
7   , ->
8     handler.addDirection( { origin: "Insert Origin Location"
9       , destination: "Insert Destination Location"})
10   return
11 )
```

Código 5: Exemplo CoffeeScript API Google-Maps-For-Rails Adaptado

Trabalhos Futuros

Trabalhos Futuros

- Para a primeira gema de exemplo **gemtranslatetoenglish**:
 - Abordar mais recursos da linguagem **Ruby**.
 - Desenvolver mais funcionalidades para mostrar mais modelos de teste.
- Para a segunda gema **Google-Maps-For-Rails adaptada**:
 - Incluir a escolha do tipo de veículo para percorrer o caminho.
 - Incluir a possibilidade de inserir pontos intermediários entre a origem e o destino.
 - Incluir a possibilidade de troca as cores entre os tracejados entre dois pontos do caminho.
 - Incluir a possibilidade de escolher o tipo de caminho a ser percorrido (mais curto, mais longo, menor tempo, maior tempo, etc).

Conclusão

Conclusão

Aprendemos:

- Conceitos sobre as bibliotecas do **Ruby**.
- Como criar uma biblioteca do **Ruby**.
- Como adaptar uma biblioteca do **Ruby**.
- Como adicionar funcionalidades em uma biblioteca do **Ruby** que mapeia a **API** do **Google Maps**.
- A utilizar a **API** do **Google Maps**.
- Conceitos sobre a linguagem **CoffeeScript**.