

WEB API - Planejamento das funções e estruturas de dados

Breno Campos, Pedro Renoir, Sávio Souza, Lucas de Amorim,
Jordão Alcântara, Gabriel de Souza.

August 2020

1 Descrição

Usaremos as bases de dados dos dados abertos de ocorrência criminais do programa SINESP do *Ministério da Justiça*. A WEB API foi modelada inicialmente para conter 18 (dezoito) requisições GET. Todas as requisições e respostas serão feitas no formato *JSON*.

1.1 GET - Número de Ocorrências por UF

O usuário deve acessar a rota `/ocorr/total/<UF:name>` onde **UF:name** é o nome do estado que se deseja.

A resposta será em formato JSON como no exemplo:

```
0 {  
1   "Bahia": 157152  
2 }
```

1.2 GET - Vítimas totais por UF

O usuário deve acessar a rota `/vit/total/<UF:name>` onde **UF:name** é o nome do estado que se deseja.

A resposta será em formato JSON como no exemplo:

```
0 {  
1   "Maranhao": 9974  
2 }
```

1.3 GET - Vítimas por Município

O usuário deve acessar a rota `/vit/<Município:name>` onde `<Município:name>` é o nome do município que se deseja.

A resposta será em formato JSON como no exemplo:

```
0 {  
1   "Aracoiaba": 43.0  
2 }
```

1.4 GET - Ocorrências de um tipo de crime no Brasil

O usuário deve acessar a rota `/ocorr/<Tipo Crime:name>` onde `<Tipo Crime:name>` é o nome do crime que se deseja pesquisar.

A resposta será em formato JSON de número inteiro como no exemplo:

```
0 266590
```

1.5 GET - Vítimas de um tipo de crime no Brasil

O usuário deve acessar a rota `/vit/<Tipo Crime:name>` onde `<Tipo Crime:name>` é o nome do crime que se deseja pesquisar.

A resposta será em formato JSON de número inteiro como no exemplo:

```
0 11348
```

1.6 GET - Vítimas por Ano

O usuário deve acessar a rota `/vit/epoc/<Ano:name>` onde `<Ano:name>` é o ano que se deseja pesquisar.

A resposta será em formato JSON de número inteiro.

```
0 11189
```

1.7 GET - Tipos de crime na tabela de Ocorrências

O usuário deve acessar a rota `/ocorr/crim/tipos`, onde retornará um array com todos os crimes desta tabela.

A resposta será em formato JSON como no exemplo:

```

0  [
1      "Estupro",
2      "Furto de veiculo",
3      "Homicidio doloso",
4      "Lesao corporal seguida de morte",
5      "Roubo a instituicao financeira",
6      "Roubo de carga",
7      "Roubo de veiculo",
8      "Roubo seguido de morte (latrocinio)",
9      "Tentativa de homicidio"
10 ]

```

1.8 GET - Tipos de crime na tabela de vítimas

O usuário deve acessar a rota **/vit/crim/tipos** onde retornará um array com todos os crimes desta tabela.

A resposta será em formato JSON como no exemplo:

```

0  [
1      "Homicidio doloso",
2      "Lesao corporal seguida de morte",
3      "Roubo seguido de morte (latrocinio)"
4  ]

```

1.9 GET - Ocorrências de um tipo de crime em um estado

O usuário deve acessar a rota **/ocorr/<Tipo Crime:name>/<UF:name>** onde **<Tipo Crime:name>** é o tipo de crime que se deseja pesquisar e **<UF:name>** é o nome do estado que se deseja pesquisa.

A resposta será em formato JSON de número inteiro como no exemplo:

```

0  3002

```

1.10 GET - Vítimas de um tipo de crime em um estado

O usuário deve acessar a rota **/vit/<Tipo Crime:name>/<UF:name>** onde **<Tipo Crime:name>** é o tipo de crime que se deseja pesquisar e **<UF:name>** é o nome do estado que se deseja pesquisa.

A resposta será em formato JSON de número inteiro como no exemplo:

0 967

1.11 GET - Vítimas de um tipo de crime por ano

O usuário deve acessar a rota `/vit/crim/epoc/<Tipo Crime:name>/<Ano:name>` onde `<Tipo Crime:name>` é o tipo de crime que se deseja pesquisar e `<Ano:name>` é o ano que se deseja pesquisa.

A resposta será em formato JSON de número inteiro como no exemplo:

0 11505

1.12 GET - Ocorrências de um tipo de crime por ano

O usuário deve acessar a rota `/ocorr/crim/epoc/<Tipo Crime:name>/<Ano:name>` onde `<Tipo Crime:name>` é o tipo de crime que se deseja pesquisar e `<Ano:name>` é o ano que se deseja pesquisa.

A resposta será em formato JSON de número inteiro como no exemplo:

0 48402

1.13 GET - Top 10 municípios com mais vítimas

O usuário deve acessar a rota `/max/munc/vit` onde retornará um array com os 10 municípios com mais vítimas e quantidade de vítimas em cada um deles.

A resposta será em formato JSON como no exemplo:

```
0 {
1   "Rio De Janeiro": 2761.0,
2   "Salvador": 2445.0,
3   "Fortaleza": 2417.0,
4   "Manaus": 1945.0,
5   "Sao Paulo": 1524.0,
6   "Belem": 1376.0,
7   "Recife": 1179.0,
8   "Porto Alegre": 1077.0,
9   "Maceio": 863.0,
10  "Belo Horizonte": 860.0
11 }
```

1.14 GET - Top 10 municípios com menos vítimas

O usuário deve acessar a rota **/min/munc/vit** onde retornará um array com os 10 municípios com menos vítimas e quantidade de vítimas em cada um deles.

A resposta será em formato JSON como no exemplo:

```
0 {  
1     "Abaira": 0.0,  
2     "Acegua": 0.0,  
3     "Adolfo": 0.0,  
4     "Afonso Cunha": 0.0,  
5     "Agricolandia": 0.0,  
6     "Aiuruoca": 0.0,  
7     "Alambari": 0.0,  
8     "Albertina": 0.0,  
9     "Alfredo Chaves": 0.0,  
10    "Alfredo Marcondes": 0.0  
11 }
```

1.15 GET - Top 10 estados com mais vítimas

O usuário deve acessar a rota **/max/estad/vit** onde retornará um array com os 10 estados com mais vítimas e quantidade de vítimas em cada um deles.

A resposta será em formato JSON como no exemplo:

```
0 {  
1     "BA": 11982.0,  
2     "RJ": 10007.0,  
3     "PE": 8155.0,  
4     "CE": 7678.0,  
5     "PA": 7069.0,  
6     "MG": 6538.0,  
7     "SP": 6492.0,  
8     "RS": 5004.0,  
9     "PR": 4312.0,  
10    "GO": 4161.0  
11 }
```

1.16 GET - Top 10 estados com menos vítimas

O usuário deve acessar a rota **/min/estad/vit** onde retornará um array com os 10 estados com menos vítimas e quantidade de vítimas em cada um deles.

A resposta será em formato JSON como no exemplo:

```

0 {
1     "RR": 372.0,
2     "AP": 603.0,
3     "RO": 766.0,
4     "AC": 779.0,
5     "TO": 815.0,
6     "DF": 923.0,
7     "MS": 1097.0,
8     "PI": 1240.0,
9     "SC": 1697.0,
10    "MT": 1901.0
11 }

```

1.17 GET - Top 10 estados com mais ocorrências

O usuário deve acessar a rota **/max/estad/ocorr** onde retornará um array com os 10 estados com mais ocorrências e quantidade de ocorrências em cada um deles.

A resposta será em formato JSON como no exemplo:

```

0 {
1     "Sao Paulo": 1010726,
2     "Rio de Janeiro": 425553,
3     "Parana": 363780,
4     "Minas Gerais": 236240,
5     "Rio Grande do Sul": 207938,
6     "Bahia": 157152,
7     "Pernambuco": 156957,
8     "Goias": 139249,
9     "Ceara": 109097,
10    "Para": 95954
11 }

```

1.18 GET - Top 10 estados com menos ocorrências

O usuário deve acessar a rota **/min/estad/ocorr** onde retornará um array com os 10 estados com menos ocorrências e quantidade de ocorrências em cada um deles.

A resposta será em formato JSON como no exemplo:

```
0 {  
1     "Acre": 8226,  
2     "Roraima": 9222,  
3     "Amapa": 10487,  
4     "Tocantins": 20243,  
5     "Paraiba": 22881,  
6     "Sergipe": 28979,  
7     "Rondonia": 33976,  
8     "Alagoas": 36792,  
9     "Mato Grosso do Sul": 39359,  
10    "Amazonas": 42323  
11 }
```