

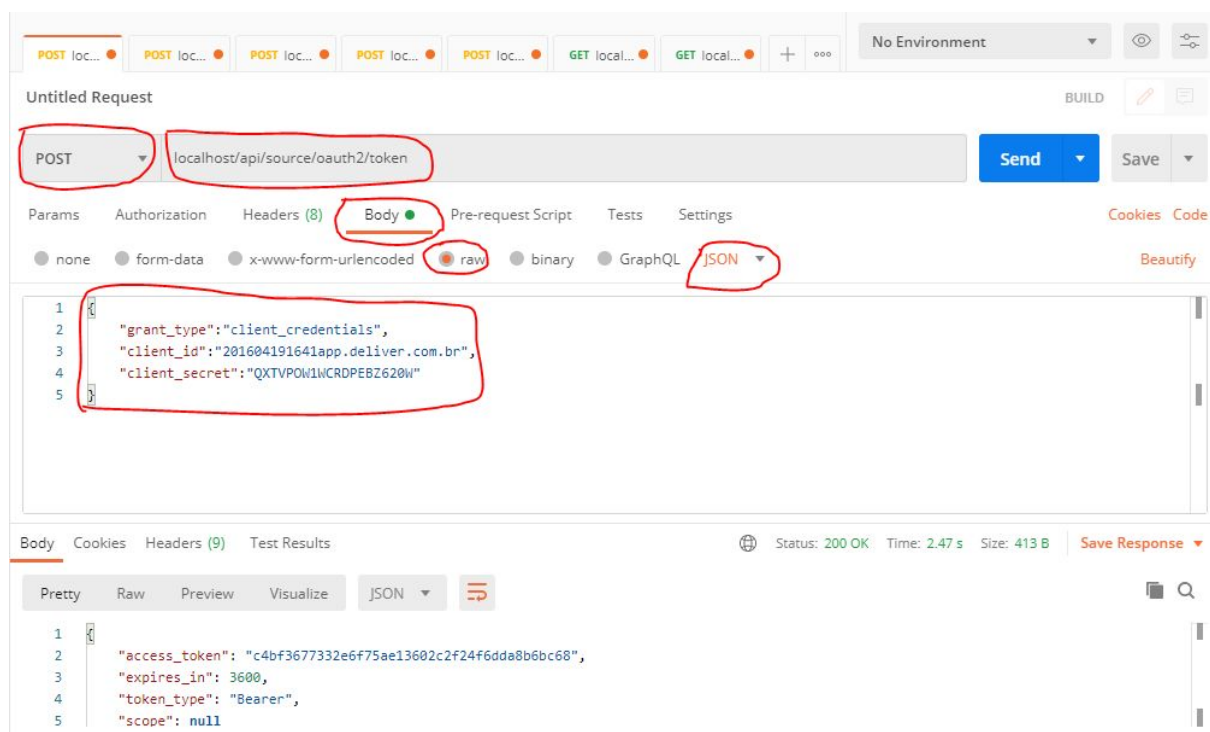
Após configurar o arquivo config.php basta executar os serviços através do software postman (<https://www.postman.com/downloads/>). Os serviços GET podem ser executados através do navegador.

As URLs dos serviços podem ser encontradas no caminho api\source\index.php

1. Este serviço serve para criar um token de autenticação para depois executar os demais serviços.

```
77:12 e remove o token  
$app->match('/oauth2/token', 'API\Core\Auth\AuthServerController::generateAccessToken');
```

No postman preencher os campos conforme mostra a imagem abaixo.



JSON que deve conter no campo conforme mostra imagem:

```
{  
  "grant_type": "client_credentials",  
  "client_id": "201604191641app.deliver.com.br",  
  "client_secret": "QXTVPOW1WCRDPEBZ620W"  
}
```

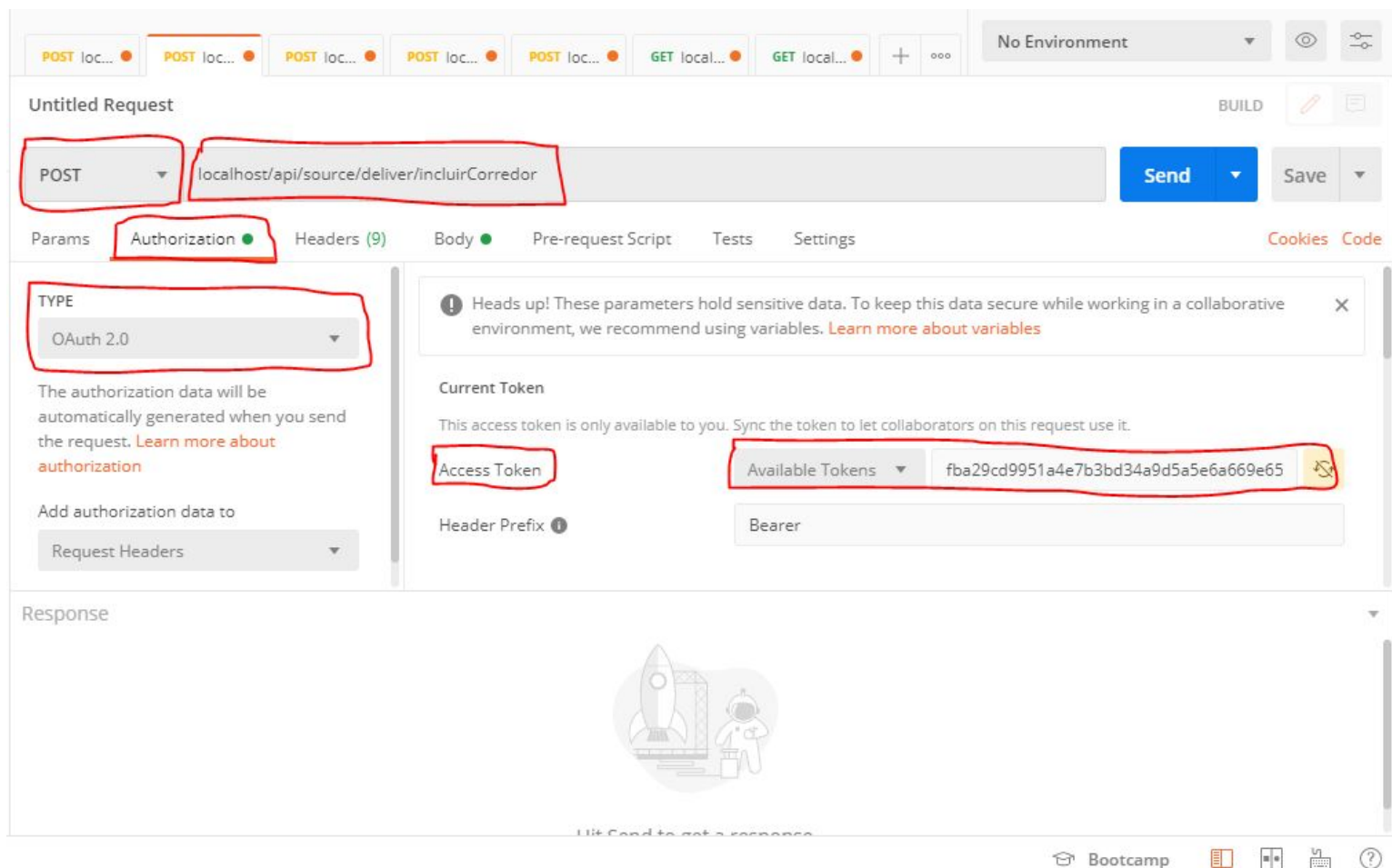
Após clicar em **send** é possível ver o token gerado pela aba pretty.



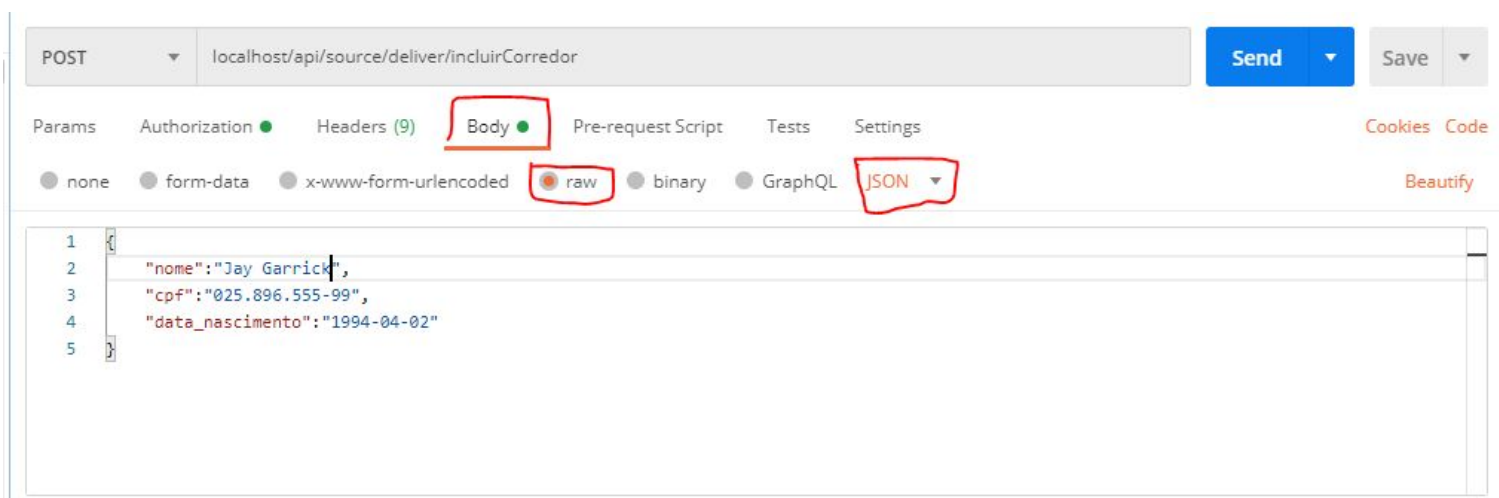
O access_token gerado deve ser usado nas demais abas do postman para execução dos serviços.

```
$app->POST('deliver/incluirCorredor','API\deliver_teste\Controller\servicoTesteController::incluirCorredor');
```

Para executar este serviço preencha os campos de acces_token conforme mostra a imagem:



Após essa etapa clicar em body => raw => JSON



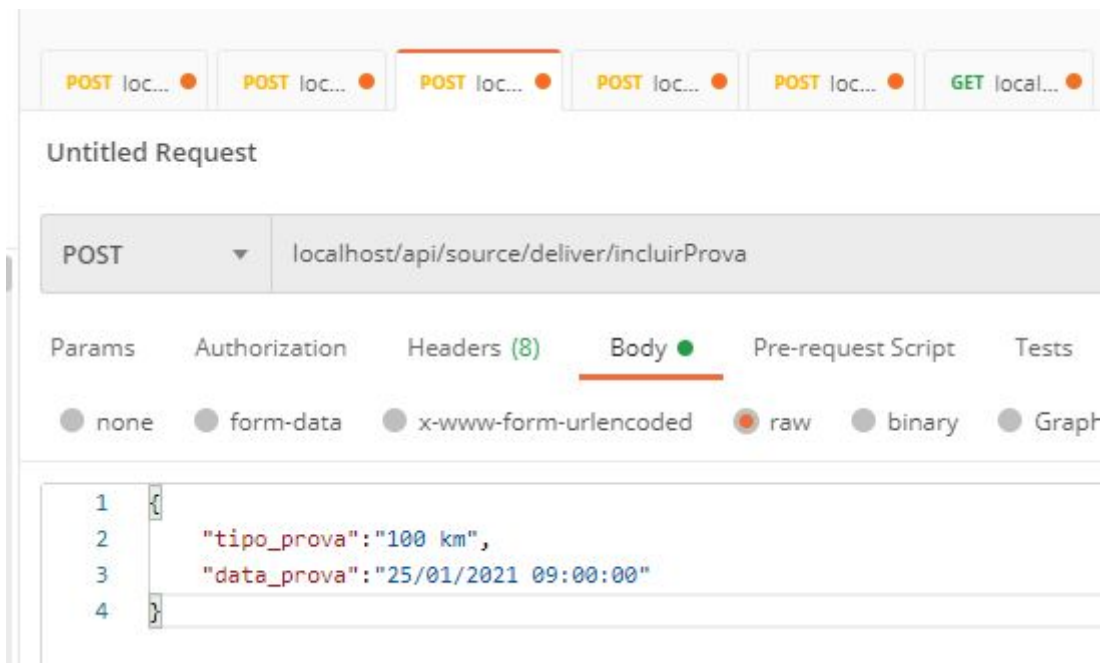
e colar a JSON abaixo

```
{
  "nome":"Jay Garrick",
  "cpf":"025.456.589-99",
  "data_nascimento":"1985-04-02"
}
```

Observação: O token expira a cada 6 minutos. Então é necessário gerar outro.

Os demais serviços de inserção seguem o mesmo padrão do listado acima. Abaixo segue os códigos JSON correspondente.

```
$app->POST('deliver/incluirProva','API\deliver_teste\Controller\servicoTesteController::incluirProva');
```



```
{
  "tipo_prova": "5 km",
  "data_prova": "24/01/2021 12:00:00"
}
```

```
$app->POST('deliver/incluirProvaCorredor','API\deliver_teste\Controller\servicoTesteController::incluirProvaCorredor');
```

POST loc... POST loc... POST loc... POST loc... POST loc...

Untitled Request

POST localhost/api/source/deliver/incluirProvaCorredor

Params Authorization Headers (8) **Body** Pre-request Scr

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ bin

```
1 {
2   "id_prova": "2",
3   "id_corredor": "6"
4 }
```

```
{
  "id_prova": "1",
  "id_corredor": "1"
}
```

```
$app->POST('deliver/incluirResultados', 'API\deliver_teste\Controller\servicoTesteController::incluirResultados')
```

POST loc... POST loc... POST loc... POST loc... POST loc...

Untitled Request

POST localhost/api/source/deliver/incluirResultados

Params Authorization Headers (8) **Body** Pre-request Scr

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ bin

```
1 {
2   "id_prova_corredor": "13",
3   "hora_inicio_prova": "09:00:00",
4   "hora_conclusao_prova": "10:00:00"
5 }
```

```
{
  "id_prova_corredor": "1",
  "hora_inicio_prova": "09:00:00",
  "hora_conclusao_prova": "11:00:00"
}
```

Para execução dos serviços de listar classificação seguir os passos para colocar o token conforme mostrado anteriormente.

Após, seguir os passos abaixo para obter o resultado da busca.

The screenshot shows the Postman interface. At the top, there's a row of request history with several POST requests and two GET requests. The main area is titled 'Untitled Request'. The method 'GET' is selected and highlighted with a red box. The URL 'localhost/api/source/deliver/listarClassificacaoPorIdade' is entered in the address bar and also highlighted with a red box. To the right are 'Send' and 'Save' buttons. Below the address bar are tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Query Params' section is visible with a table:

KEY	VALUE	DESCRIPTION
Key	Value	Description

Below this, the 'Body' tab is selected and highlighted with a red box. It shows the response in 'Pretty' format, also highlighted with a red box. The response is a JSON object:

```
{
  "5 km": {
    "26-35": [
      {
        "id_prova": "1",
        "tipo_prova": "5 km",
        "id_corredor": "2",
        "idade": "27",
        "nome": "Barry Allen",
        "classificacao": 1
      }
    ]
  }
}
```

At the top right, it says 'No Environment'. At the bottom right, it shows 'Status: 200 OK', 'Time: 11.77 s', 'Size: 1.64 KB', and a 'Save Response' button.

A url também pode ser colocada no navegador como mostra abaixo:

localhost/api/source/deliver/lista x +
localhost/api/source/deliver/listarClassificacaoPorIdade

```
{
  "5 km": {
    "26-35": [
      {
        "id_prova": "1",
        "tipo_prova": "5 km",
        "id_corredor": "2",
        "idade": "27",
        "nome": "Barry A",
        "id_corredor": "4",
        "idade": "31",
        "nome": "Eobard Thawne",
        "classificacao": 2
      },
      {
        "id_prova": "1",
        "tipo_prova": "5 km",
        "id_corredor": "5",
        "idade": "38",
        "nome": "Jay Garrick",
        "classificacao": 1
      },
      {
        "id_prova": "1",
        "tipo_prova": "5 km",
        "id_corredor": "6",
        "idade": "23",
        "nome": "Jessy quick",
        "classificacao": 1
      },
      {
        "id_prova": "1",
        "tipo_prova": "5 km",
        "id_corredor": "6",
        "idade": "23",
        "nome": "Jessy quick",
        "classificacao": 2
      }
    ],
    "18-25": {
      "2": {
        "id_prova": "2",
        "tipo_prova": "10 km",
        "id_corredor": "1",
        "idade": "21",
        "nome": "Wally",
        "id_corredor": "6",
        "idade": "23",
        "nome": "Jessy quick",
        "classificacao": 2
      },
      "26-35": {
        "2": {
          "id_prova": "2",
          "tipo_prova": "10 km",
          "id_corredor": "4",
          "idade": "31",
          "nome": "Jay Garrick",
          "classificacao": 1
        },
        "3": {
          "id_prova": "2",
          "tipo_prova": "10 km",
          "id_corredor": "3",
          "idade": "41",
          "nome": "Jay Garrick",
          "classificacao": 1
        },
        "3": {
          "id_prova": "2",
          "tipo_prova": "10 km",
          "id_corredor": "3",
          "idade": "41",
          "nome": "Jay Garrick",
          "classificacao": 1
        }
      }
    }
  }
}
```

Os mesmos passos podem ser realizados para executar o serviço de listar classificação geral

```
$app->GET('deliver/listarClassificacaoGeral','API\deliver_teste\Controller\servicoTesteController::listarClassificacaoGeral');
```