

TXT Credenciamento

Guia para Desenvolvimento de API

| | |
|--|----------|
| 1 - Estrutura de dados no TXT Credenciamento..... | 1 |
| 1.1 - Modelo da Credencial..... | 1 |
| 1.2 - Tipo da Credencial..... | 1 |
| 1.3 - Status da Credencial..... | 1 |
| 1.4 - Numeração..... | 2 |
| 1.5 - Código..... | 2 |
| 1.6 - Foto..... | 2 |
| 1.7 - Áreas..... | 2 |
| 1.7.1 - Áreas padrões..... | 2 |
| 2 - Integração..... | 3 |
| 2.1 - APIs REST..... | 3 |
| 2.2 - Autenticação..... | 3 |
| 2.3 - Importação das credenciais..... | 3 |
| 2.3.1 - Filtro por data de modificação..... | 3 |
| 2.3.2 - Paginação do resultado..... | 4 |
| 2.3.3 - Dados das credenciais..... | 4 |
| 2.3.4 - Fotos..... | 5 |
| 2.4 - Envio dos acessos..... | 5 |
| 3 - Exemplo de API..... | 6 |
| 3.1 - Busca de credenciais..... | 6 |
| 3.2 - Download da foto..... | 7 |
| 3.3 - Registro de acesso..... | 8 |
| 4 - Requisitos para faces..... | 8 |
| 4.1 - Dimensões da imagem..... | 9 |
| 4.2 - Critérios de cadastro..... | 9 |

1 - Estrutura de dados no TXT Credenciamento

O TXT Credenciamento possui uma estrutura de dados que permite a personalização para variados tipos de eventos. Sendo assim é importante entender essa estrutura para facilitar a importação dos dados das credenciais.

1.1 - Modelo da Credencial

O modelo da credencial define os campos (com suas respectivas validações) que as credenciais vão utilizar. É ele que define o formulário que será preenchido e quais campos poderão ser impressos na credencial.

Normalmente é configurado apenas um modelo de credencial, porém alguns eventos exigem diferenciação dos modelos. Por exemplo:

- **Modelo 1: Funcionário**
 - **Campos:** Nome, RG, CPF, Data de Nascimento, Departamento, Função
- **Modelo 2: Visitante**
 - **Campos:** Nome, Empresa, CPF

Essa divisão de modelos deve ser usada apenas em casos realmente necessários, tendo em vista que pode aumentar a complexidade na operação e na utilização de alguns módulos.

1.2 - Tipo da Credencial

O tipo da credencial define as propriedades da credencial:

- **Modelo da credencial**
- **Tipo de numeração** (manual ou [numerador](#))
- **Tipo de código** (manual ou [codificador](#))
- **Layout de impressão**
- **Foto** (se utiliza [foto](#), sua obrigatoriedade e a proporção de recorte - ex: 3x4, 5x7)
- **Acessos padrões** ([opcional](#))

Os tipos sempre pertencem a um modelo de credencial. Caso haja um tipo com o mesmo nome em modelos diferentes eles são cadastrados como tipos diferentes, um em cada modelo.

1.3 - Status da Credencial

Os status da credencial dentro do TXT Credenciamento são:

- **Solicitado:** credencial solicitada através do Cred Net
- **Pendente:** credencial pendente de impressão
- **Aprovado:** credencial solicitada foi aprovada
- **Recusado:** credencial solicitada foi reprovada
- **Ativo:** credencia impressa / em uso
- **Utilizado:** credencial já utilizada no acesso ao menos uma vez
- **Cancelado:** credencial cancelada

1.4 - Numeração

O número da credencial visa facilitar sua identificação através de uma representação de 64 bits (de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807).

Normalmente se usa uma numeração sequencial a partir do número 1, mas este campo pode ficar aberto para digitação (ou importação) e não possui validação para números repetidos por se tratar apenas de um código de referência.

Quando se usa numeração automática é possível definir um número inicial e o valor de incremento. Esses numeradores são definidos por tipo de credencial.

1.5 - Código

O código da credencial é utilizado para identificar a credencial no controle de acesso através de uma representação alfanumérica de 50 caracteres.

Normalmente este código é gerado por um codificador padrão do sistema, mas é possível configurar novos codificadores, com valores de início, incremento, prefixo e sufixo personalizados. Também é possível criar novos codificadores com algoritmos exclusivos (*inclui custos de desenvolvimento*).

Também é possível deixar o campo código aberto para digitação (ou importação), porém este possui validação e não permite o preenchimento de códigos duplicados.

O tipo de codificação também é definido por tipo de credencial.

1.6 - Foto

O uso de foto na credencial é configurado em seu tipo. Quando um tipo de credencial possui foto é necessário definir a proporção da foto.

A proporção da foto é usada apenas nos módulos TXT Credenciamento e TXT Cred.Net para realizar o recorte automático no tamanho que permita a sua impressão na credencial sem distorções. Esse recorte automático não é realizado nas importações e integrações.

1.7 - Áreas

As áreas de acesso servem para identificar quais recursos (setores, seções, permissões, etc) a credencial pode ter acesso.

Todas as áreas cadastradas para o evento ficam disponíveis para todos os modelos e tipos de credenciais. Elas podem ser divididas por grupos para facilitar a visualização e atribuição no TXT Credenciamento.

As credenciais possuem um valor booleano (*verdadeiro ou falso*) para todas as áreas configuradas no evento. É através desses valores que os módulos de controle de acesso identificam se a credencial pode acessar um recurso.

1.7.1 - Áreas padrões

É possível definir nos tipos de credenciais um modelo padrão de permissão de acesso, determinando quais áreas o tipo de credencial tem acesso. Além disso, é possível definir quais áreas de acesso podem ser alteradas.

Quando uma área não é definida no modelo padrão, ela pode ser adicionada para uma credencial. Quando uma área é definida como bloqueada no modelo padrão, a área não pode ser adicionada para uma credencial.

2 - Integração

Durante o briefing do projeto é definido se o credenciamento utilizará servidor de banco de dados na nuvem ou local. Normalmente é utilizado banco de dados em servidor local para evitar indisponibilidade em caso de problemas com o link de internet. Sendo assim o TXT Credenciamento não possui APIs genéricas para integração, já que nem sempre o servidor local possui rota externa para outros sistemas disponíveis.

Como as integrações são desenvolvidas de forma personalizada, consumindo APIs dos parceiros, não existe um modelo padrão. Porém existem algumas regras e sugestões de como essa integração deve ser feita.

2.1 - APIs REST

É recomendado o uso de APIs REST com o payload em formato JSON, devido à padrões definidos de mercado (preferencialmente aderindo ao padrão *RESTful*).

Apesar da possibilidade de usar outros padrões, a complexidade pode afetar diretamente o custo de desenvolvimento.

2.2 - Autenticação

Normalmente é utilizado um token fixo, informado através de um header na requisição HTTP. Essa abordagem simplifica a integração e o diagnóstico de problemas.

É possível usar outros tipos de autenticação, inclusive com tempo de expiração. Porém estes podem aumentar o tempo de resposta e os pontos de falha, além da complexidade no desenvolvimento também.

2.3 - Importação das credenciais

A importação das credenciais é o fluxo em que o integrador inclui, atualiza e exclui as credenciais no TXT Credenciamento.

Para realizar esta operação o integrador necessita de uma API que retorne os dados das credenciais que precisam ser atualizadas. Como o volume de credenciais pode ser grande, é recomendável o uso de parâmetros que permitam o filtro da busca e a paginação do resultado.

2.3.1 - Filtro por data de modificação

A integração é realizada através de consultas na API do parceiro, em intervalos pré configurados, para obter as informações das credenciais que precisam ser atualizadas no TXT Credenciamento. Por isso é recomendável usar um filtro que permita essa consulta de maneira otimizada, e para isso normalmente utilizamos a data de alteração da credencial.

Através da data de última atualização (esta pode ser a data da inclusão no caso de uma nova credencial) podemos buscar somente as credenciais necessárias. É recomendável que, além da utilização no filtro, essa data também seja retornada junto com as informações da credencial, possibilitando assim o uso da maior data para as buscas

seguintes. Desta forma usamos sempre a data e hora seguindo com o GMT configurado no servidor do parceiro.

Quanto mais preciso estiver o formato, mais eficiente será a busca e evitará a carga de dados já processados. É recomendável o envio desta data via *query parameter*, geralmente em formato ISO (ex.: 2023-12-31T23:59:59.999) ou em formato UNIX Timestamp (ex.: 1704077999999).

2.3.2 - Paginação do resultado

Para tornar a consulta mais performática, em casos onde API pode retornar um grande volume de dados, é recomendável paginar o resultado.

Assim o retorno pode usar uma estrutura onde possui metadados como a quantidade total de registros, quantidade de registros por página, número da página, número total de páginas, etc. O importante é possuir dados suficientes para realizar todas as consultas necessárias para obter todos os registros.

Existem também casos onde o volume de dados das credenciais é pequeno e trazer o resultado sem paginação pode simplificar a integração.

Exemplo de estrutura de paginação:

```
JavaScript
{
  "pageCount": 500,
  "totalItemCount": 5000,
  "pageNumber": 2,
  "pageSize": 10,
  "hasPreviousPage": true,
  "hasNextPage": true,
  "isFirstPage": false,
  "isLastPage": false,
  "items":
  []
}
```

2.3.3 - Dados das credenciais

O resultado da busca precisa retornar todos os dados das credenciais que serão importados. O modelo e tipo da credencial são campos que podem ser omitidos caso seja utilizado somente um. O número e código das credenciais são campos obrigatórios caso o tipo não use numerador e codificador respectivamente. O status da credencial pode ser um campo definindo se a credencial está ativa ou cancelada.

Além dos campos que serão importados para o TXT Credenciamento, as credenciais precisam ter um identificador único para poder realizar as atualizações em credenciais já existentes. Este campo pode ser o próprio código, desde que não haja duplicidade mesmo em credenciais excluídas.

Um campo para identificar a importação da foto também pode ser retornado, ou dependendo do caso o arquivo pode ser serializado em Base 64.

O campo data de alteração ajuda a tornar a importação mais eficiente, pois pode ser usado como parâmetro para as consultas subsequentes.

Um campo para identificar se a foto foi ou não atualizada também ajuda na performance da importação (evita consultas e atualizações desnecessárias nos arquivos de foto).

Campos booleanos ou lista podem ser utilizados para identificar as áreas de acesso da credencial.

Um exemplo de item:

```
JavaScript
{
  "credentialId": 384,
  "number": 1004,
  "code": "80763777458",
  "type": "VIP",
  "active": true,
  "name": "John Hausmann",
  "document": "29291-3",
  "photo": "filename.jpg",
  "areas":
  [
    "SETOR_1",
    "SETOR_2"
  ]
}
```

2.3.4 - Fotos

As fotos das credenciais podem ser fornecidas junto com os dados da credencial, em um campo texto com arquivo serializado em Base 64. Esta maneira é mais simples por não exigir mais de uma chamada, porém é preciso tomar cuidado com o tamanho da resposta, limitando a quantidade de registros e o tamanho das fotos.

Outra forma é disponibilizar outra API para obter o arquivo, sendo parametrizado o endpoint informando um identificador da credencial ou um campo no item. Também é possível disponibilizar em um dos campos a URL pré assinada para o download do arquivo.

As fotos disponibilizadas pela API precisam estar em formato PNG ou JPG e precisam estar tratadas, pois são gravadas no TXT Credenciamento da forma que são obtidas. Validações, redução de tamanho e recortes devem ser realizados nas fotos antes de serem disponibilizadas para importação.

Para utilização das fotos em controle de acesso facial as fotos devem seguir os requisitos de faces.

2.4 - Envio dos acessos

É possível disponibilizar uma API para o integrador enviar o registro de leituras e acessos realizados nos módulos de controle de acesso.

Normalmente os campos enviados são:

- Id ou código da Credencial
- Identificador da área acessada
- Ação realizada (entrada / saída)
- Data e Hora da ação

Exemplo de registro de acesso:

```
JavaScript
{
  "code": "80763777458",
  "area": "SETOR_1",
  "action": "ENTRY",
  "action_date": "2023-01-01T13:38:02.362"
}
```

3 - Exemplo de API

Um exemplo de integração com fotos e controle de acesso.

3.1 - Busca de credenciais

| | |
|-----------------|---|
| Method | GET |
| URL | http://exemple.com/credentials/search |
| Params | - date_from : 2020-01-01T00:00:00.000 |
| Headers | - authorization : l9aYK722rPO3gCoFK/xsOw== |
| Request | <pre>curl --request GET \ --url 'http://exemple.com/credentials/search?date_from=2020-01-01T00%3A00%3A00.000' \ --header 'authorization: l9aYK722rPO3gCoFK/xsOw=='</pre> |
| Response | <p>HTTP/1.1 200 OK Content-Type: application/json; charset=utf-8</p> <pre>JavaScript { "pageCount": 1, "total": 2, "pageNumber": 1, "pageSize": 100, "hasNextPage": false, "items":</pre> |

```

[
  {
    "credentialId": 384,
    "number": 1004,
    "code": "80763777458",
    "type": "VIP",
    "active": true,
    "name": "John Hausmann",
    "document": "29291-3",
    "areas":
    [
      "SETOR_1",
      "SETOR_2"
    ]
  },
  {
    "credentialId": 385,
    "number": 1004,
    "code": "87745807637",
    "type": "Staff",
    "name": "Nelson Noble",
    "document": "32342-8",
    "areas":
    [
    ]
  }
]

```

3.2 - Download da foto

| | |
|-----------------|--|
| Method | GET |
| URL | http://exemple.com/credentials/photo |
| Params | - credential_id : 384 |
| Headers | - authorization : l9aYK722rPO3gCoFK/xsOw== |
| Request | curl --request GET \ --url 'http://exemple.com/credentials/photo?credential_id=384' \ --header 'authorization: l9aYK722rPO3gCoFK/xsOw==' |
| Response | HTTP/1.1 200 OK Content-Type: image/jpeg |

| | |
|--|---|
| | <pre>JavaScript <<Image Bytes>></pre> |
|--|---|

3.3 - Registro de acesso

| | |
|-----------------|---|
| Method | POST |
| URL | http://exemplo.com/acess/register |
| Headers | <ul style="list-style-type: none"> - authorization: l9aYK722rPO3gCoFK/xsOw== - Content-Type: application/json |
| Request | <pre>curl --request POST \ --url 'http://exemplo.com/acess/register?date_from=2020-01-01T00%3A00%3A00.000' \ --header 'Content-Type: application/json' \ --header 'authorization: l9aYK722rPO3gCoFK/xsOw==' \ --data ' JavaScript { "code": "80763777458", "area": "SETOR_1", "action": "ENTRY", "action_date": "2023-01-01T13:38:02.362" } '</pre> |
| Response | HTTP/1.1 202 OK |

4 - Requisitos para faces

No cadastro de fotos para reconhecimento facial, são aceitas imagens de rosto nos formatos JPG ou PNG. É importante que as fotos utilizadas para cadastro contenham apenas um rosto e que este tenha uma boa iluminação, distribuída uniformemente.

As fotos devem ser de pessoas com todo o rosto claramente visível na foto. Além disso, as pessoas devem estar olhando diretamente para a câmera. O uso de máscara no cadastro é permitido, mas não é recomendado. O uso de óculos de grau é perfeitamente aceitável, mas o uso de óculos escuros não é.

4.1 - Dimensões da imagem

Serão aceitas imagens com tamanho mínimo de 160x160 pixels. O limite superior é 1920 x 1080 = 2.073.600 pixels. Assim, por exemplo, imagens de 300x300, 600x600, 2000x1000 serão aceitas mas 4000x3000 não.

4.2 - Critérios de cadastro

Há ainda outros critérios que precisam ser observados em relação à qualidade da face contida na foto para cadastro:

- **Largura da face na foto:** A foto deve conter uma face que não esteja nem muito próxima nem muito distante. A região da face deve estar totalmente contida na foto, sendo mantida uma pequena margem em relação a cada um dos lados.
- **Centralização da face:** A região da face deve ocupar o centro da imagem tanto na direção vertical quanto na horizontal.
- **Centralização da pose:** A pessoa na foto deve estar com a face voltada para a câmera, olhando para a frente.
- **Nitidez da região da face:** A região da face deve estar nítida e bem iluminada.

Para mais detalhes sobre os critérios para cadastro de faces por foto:

<https://www.controlid.com.br/manual/idface-cadastramento-de-foto.pdf>