

Lista de Serviços para Aplicativo do Acionamento 193/VASB

1º Serviço GET – Confirmar Cadastro

Endpoint: <http://srvhomolog01:8080/SRVAPPCCB/api/Credenciamento/ConfirmarCadastro/{email}>

Descrição: Serviço valida através do e-mail o cadastro do VASB no sistema de credenciamento.

Parâmetros via QueryString: e-mail

Verbo HTTP: GET

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 - OK

400 - E-mail Inválido

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada: parâmetro passado via URI

JSON Resposta:

```
{
  "codigo": 0/1
  "descricao": Inexistente/Existente
}
```

2º Serviço POST – Enviar Token

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/CobomOnline/EnviarToken/>

Descrição: Enviar Token para usuário que não possui credenciamento (Não VASB). O token será a senha de acesso do usuário.

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 – Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  "nmPes": varchar(200)  Nome
  "tipoDoc": tinyint      Documento
  "numCpf": varchar(30)   CPF
  "numRg": varchar(30)    RG
  "ddd": tinyint          DDD
  "numTel": int            Telefone
  "email": varchar(20)    E-mail
}
```

JSON Resposta:

```
{
  "idPes": bigint        Identificação da Pessoa
}
```

3º Serviço POST – Validar Token

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/CobomOnline/ValidarToken/>

Descrição: Validar Token inserido pelo usuário (Não VASB).

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 – Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  idPes: bigint
  email: varchar(200)
  idCodToken: int      Token
}
```

JSON Resposta: N/A

4º Serviço POST – Autenticar Usuário Credenciado

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/Credenciamento/AutenticarUsuarioCredenciado>

Descrição: Autenticar credenciais do usuário acesso ao app.

Parâmetros via QueryString: N/A.

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 - Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  email: varchar(50)
  senha: varchar(100) - Usar Encoding: "utf-8" Criptografia: Hash / Formato dos valores: "MD5" e "SHA1"
}
```

JSON Resposta:

```
{
  idPes: bigint
}
```

5º Serviço GET – Verificar Credenciamento Ativo

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/Credenciamento/VerificarCredenciamentoAtivo/{idPes}>

Descrição: Serviço deverá validar se o credenciamento do VASB consta ativo no sistema.

Parâmetros via QueryString: idPes

Verbo HTTP: GET

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 - OK

400 – Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada: parâmetro passado via URI

JSON Resposta:

```
{
  status: int      Status
  descSolic: varchar(3500)  Descrição da solicitação
}
```

6º Serviço GET – Consultar Árvore de Decisão por Natureza

Endpoint: <SRVAPPCCB/api/Credenciamento/ConsultarArvoreDecisaoNatureza/{NcdCod}>

Descrição: Serviço irá consultar respostas de perguntas parametrizados feita pelo usuário ao abrir uma solicitação de acordo com a natureza da ocorrência.

Parâmetros via QueryString: NcdCod

Verbo HTTP: GET

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 - OK

400 - E-mail Inválido

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada: parâmetro passado via URI

JSON Resposta:

```
{
  idQuest: tinyint  Identificação do questionário
  {
    idPrg: smallint  Identificação da pergunta
    nmPrg: varchar(300) Pergunta
    {
      idTipResp: smallint  Identificação do tipo de resposta
    }
  }
}
```

```

nmRespMlpEsc: varchar(300) Nome das respostas de múltipla escolha
descRespCta: varchar(256) Descrição da resposta curta
    }
}
}

```

Natureza {NcdCod}: **623** – Incêndio outros/ **627** – Incêndio em vegetação / **629** - Incêndio em veículo / **937** – Vítima de acidente de trânsito/Atropelamento / **943** – Afogamento em curso / **959** – Vítima Inconsciente (Parada Cardiorrespiratória) / **962** – Incêndio em edificação / **977** - Engasgamento

7º Serviço POST – Abrir Solicitação

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/CobomOnline/AbrirSolicitacao>

Descrição: Possibilitar o usuário de abrir solicitações (Ocorrência).

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 - E-mail Inválido

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
```

idPes: bigint

logradouro: varchar(200) **Logradouro**

num: int **Número**

bairro: varchar(100) **Bairro**

ltd: float **Latitude**

lng: float **Longitude**

codNat: char(3) **Código da natureza**

```
{
```

idQuest: tinyint **Identificação do questionário**

idPrg: smallint **Identificação da pergunta**

descPrg: varchar(300) **Descrição da pergunta**

```
{
```

tpResp: smallint **Tipo de resposta**

nmDescResp: varchar(30) **Nome da descrição da resposta**

idResUsuMtpEsc: smallint **Identificação da resposta múltipla escolha do usuário**

nmDescRespMtpEsc: varchar(300) **Nome da descrição múltipla escolha**

idResUsuRespCta: smallint **Identificação da resposta curta do usuário**

```

        descRespCta: varchar(256) Descrição da resposta curta
    }
}

foto: varchar(300)
}

JSON Resposta:
{
    numSolic: int Número da solicitação
    dataSolic: date Data da solicitação
}

```

8º Serviço GET – Acompanhar Status Solicitação

Endpoint:

<http://srvhomolog01:8080/SRVAPPCCB/api/CobomOnline/AcompanharStatusSolicitacao/{numSolic}/{dataSolic}>

Descrição: Possibilitar o usuário de acompanhar o status da solicitação aberta via app.

Parâmetros via QueryString: numSolic e dataSolic

Verbo HTTP: GET

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK
 400 – Campos Obrigatórios Inválidos
 400 - Erro interno
 404 - Serviço Indisponível

JSON Entrada: parâmetro passado via URI

JSON Resposta:

```

{
    numSolic: int Número da solicitação
    idSitAtd: tinyint Identificação situação do atendimento
    descSitAtd: varchar(20) Situação do atendimento
    dataHoraSitAtd: datetime
}

```

9º Serviço POST – Informar Disponibilidade

Endpoint:

<http://srvhomolog01:8080/SrvAppCCB/api/CobomOnline/InformarDisponibilidade>

Descrição: Serviço que informa a disponibilidade do VASB. Caso 1- Disponível, consumir de 5 em 5 minutos para informar geolocalização do VASB

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 - Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  idPes: bigint
  idSitVasb: tinyint (1 – Disponível e 2 – Indisponível) Identificação da situação do VASB
  geoLocal: geography Geolocalização da disponibilidade
}
```

JSON Resposta: N/A

11º Serviço POST – Rastrear Atendimento

Endpoint:

<http://srvhomolog01:8080/SrvAppCCB/api/CobomOnline/RastrearAtendimento>

Descrição: Rastrear o status do VASB desde o despacho até o aceite, recusa e cancelamento do atendimento da ocorrência e caso haja o aceite e iniciada rastrear de 1 em 1 minuto o status do VASB.

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 - Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  idPes: bigint
  numOcr: int Número da ocorrência
  datOcr: datetime Data da ocorrência
  idCad: smallint Identificação do CAD
  idSitVasb: tinyint ( 1- Despachado/2-Aceita/ 3 – Iniciada / 4-Negada / 5-Cancelada: Desiste do
    Aceite)
  ltd: float Latitude
  lng: float Longitude
}
```

JSON Resposta: N/A

12º Serviço POST – Iniciar/Finalizar Atendimento

Endpoint:

<http://srvhomolog01:8080/SrvAppCCB/api/CobomOnline/IniciarFinalizarAtendimento>

Descrição: Serviço valida status de atendimento do VASB e registra quando o voluntário inicia e finaliza atendimento.

Parâmetros via QueryString: N/A

Verbo HTTP: POST

Cabeçalhos HTTP: Content-Type: application/json; Access-Control-Allow-Origin: *.

Authorization: Basic <<hash criptografada>>

Status HTTP Esperados:

200 – OK

400 - Campos Obrigatórios Inválidos

400 - Erro interno

404 - Serviço Indisponível

JSON Entrada:

```
{
  idPes: bigint
  numOcr: int
  ltd: float
  lng: float
  idSitVasb: tinyint (3- Iniciada / 2- Finalizada) Situação do VASB
  confirmaOcr: bit (Se confirmado a ocorrência é obrigatório o preenchimento dos campos numQtdVit e numNecAp)
  numQtdVit: smallint Quantidade de vítimas
  numNecAp: bit Necessita apoio
  nmTexDesc: varchar(300) Observação no atendimento
  foto: varchar(300)
}
```

JSON Resposta: N/A