



**AVISTA S/A ADMINISTRADORA DE CARTOES DE CRÉDITO
NGI – NÚCLEO DE GESTÃO DA INFORMAÇÃO**

DOCUMENTAÇÃO DO WEBSERVICE PARA O APLICATIVO “FAZER CARTÃO”

Vitória, 15 Outubro de 2014.

HISTÓRICO DE REVISÕES

Versão	Autores	Descrição da Versão	Data
1.0	Rodrigo Tosta Trancoso	Versão Inicial	03/11/2014

VISÃO GERAL

Este documento visa apresentar o funcionamento do WebService para o aplicativo mobile “fazer cartão”. Serão demonstrados os métodos disponibilizados pelo WebService.

Vale ressaltar que os objetivos deste documento são voltados para o analista/programador e também para a empresa responsável pelo aplicativo. Dentre os objetivos, os principais são:

- Explicar a arquitetura em que o WebService foi desenvolvido;
- Apresentar a forma de consumir o mesmo;
- Apresentar a forma de usar os dados retornados pelo mesmo;

Sumário

1 - Introdução	4
1.1 - JSON.....	4
1.1.1 - Objeto	4
1.1.2 - Array	5
2 - Funcionalidades	6
2.1 - Cadastro de Celular	6
2.2 - Consultar CPF	7
2.3 - Consultar CEP	8
2.4 - Consultar Escolaridade	9
2.5 - Consultar Estado Civil	10
2.6 - Cadastrar	11
2.6.1 - Pré Pago	12
2.6.2 - Pós Pago.....	12
2.7 – Enviar Documentos	13
2.8 - Consultar Datas de Vencimento	13
2.9 - Configurar Limite Crédito	14
Apêndice A.....	16

1 - Introdução

O WebService foi desenvolvido para ser consumido pelo aplicativo mobile “fazer cartão”. Para o desenvolvimento do mesmo, foi definida a utilização da tecnologia RESTFull, essa tecnologia possibilita transformar uma aplicação web em um WebService, e possibilita tanto consumir quanto produzir dados na forma de XML e de JSON, a mesma também suporta todos os métodos de requisições HTTP (GET, POST, PUT, DELETE), no entanto foi definido a utilização do JSON como estrutura de tráfego e que todas as requisições utilizarão o método POST.

1.1 - JSON

JSON é um formato leve para transmissão de dados computacionais, é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript propriamente dito.

Um dos principais motivos pela escolha do JSON é o fato de ser muito mais fácil escrever, ler e analisar as estruturas montadas.

Em JSON, podemos apresentar os dados da seguinte forma:

- Objeto;
- Array;

1.1.1 - Objeto

Um objeto JSON é um conjunto desordenado de pares atributo/valor. Um objeto começa com “{” (chave de abertura) e termina com “}” (chave de fechamento). Cada nome é seguido por “:” (dois pontos) e os pares nome/valor são seguidos por “,” (vírgula).

No exemplo abaixo, temos um objeto em JSON que contém uma estrutura capaz de armazenar os dados de um determinado funcionário.

```
1 {  
2   "NOME": "RODRIGO",  
3   "SOBRENOME": "TOSTA TRANCOSO",  
4   "IDADE": "22 ANOS",  
5   "EMPRESA": "AVISTA",  
6   "CARGO": "PROGRAMADOR"  
7 }
```

Observe que a estrutura contém a chave de abertura, os dois pontos que separam o nome do atributo e seu valor, a vírgula para separar os conjuntos de atributo/valor e por fim, a chave de fechamento.

1.1.2 - Array

Um array é uma coleção de valores ou de objetos. O array começa com "[" (colchete de abertura) e termina com "]" (colchete de fechamento). Os seus valores ou objetos também são separados por "," (vírgula).

No exemplo abaixo, temos um array em JSON que contém uma estrutura capaz de armazenar uma lista de funcionários.

```
1 {  
2   "FUNCIONÁRIOS": [  
3     {  
4       "NOME": "RODRIGO",  
5       "SOBRENOME": "TRANCOSO",  
6       "IDADE": "22 ANOS",  
7       "EMPRESA": "AVISTA",  
8       "CARGO": "PROGRAMADOR"  
9     },  
10    {  
11      "NOME": "CICLANO",  
12      "SOBRENOME": "DE ALGUMA COISA",  
13      "IDADE": "27 ANOS",  
14      "EMPRESA": "AVISTA",  
15      "CARGO": "PROGRAMADOR"  
16    },  
17    {  
18      "NOME": "BELTRANO",  
19      "SOBRENOME": "DE OUTRA COISA",  
20      "IDADE": "31 ANOS",  
21      "EMPRESA": "AVISTA",  
22      "CARGO": "PROGRAMADOR"  
23    }  
24  ]  
25 }
```

Observe que a estrutura contém a chave de abertura, os dois pontos que separam o nome do atributo e seu valor, o colchete de abertura, a vírgula para separar os conjuntos de atributo/valor, o colchete de fechamento e por fim, a chave de fechamento.

2 - Funcionalidades

O WebService disponibiliza funcionalidades para o acesso e consumo do aplicativo. As funcionalidades são:

- Cadastro de Celular;
- Consultar CPF;
- Consultar CEP;
- Consultar Escolaridades;
- Consultar Estado Civil;
- Cadastrar;
- Enviar Documentos;
- Consultar Dias de Vencimento;
- Configurar Limite Crédito;

Vale ressaltar que cada uma destas funcionalidades é acessada através de requisições HTTP, onde cada uma possui um endereço (URL) específico e único.

2.1 - Cadastro de Celular

A funcionalidade “cadastro de celular” realiza a primeira etapa do cadastro.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/cadastrar/celular/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número de celular do cliente, para que o mesmo possa ser cadastrado em nossa base de dados. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro de celular.

```
1 {  
2   "PARAM_NUMERO_CELULAR": "27999437851"  
3 }
```

Vale ressaltar que o número de telefone pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Outro ponto a ressaltar é que a validação do código segurança será realizada no próprio aplicativo, e o mesmo será devolvido no JSON de retorno.

Quando o cadastro foi bem sucedido, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": {
3     "CODIGO": "70710"
4   },
5   "RETORNO": 0,
6   "MENSAGEM": "O número de celular que você informou precisa estar
em suas mãos, pois você receberá um SMS com o código de segurança
para finalizar o seu cadastro."
7 }

```

No caso de falha no cadastro, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.2 - Consultar CPF

A funcionalidade “consultar CPF” tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CPF informado.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/consultar/cpf/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o CPF do cliente, para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CPF.

```

1 {
2   "PARAM_NUMERO_CPF": "13805236727"
3 }

```

Vale ressaltar que o número de CPF pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": {
3     "DATA_NASCIMENTO": "19/09/1992",
4     "NOME": "RODRIGO TOSTA TRANCOSO"
5   },
6   "RETORNO": 0
7 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.3 - Consultar CEP

A funcionalidade “consultar CEP” tem a finalidade, como o próprio nome já diz, de devolver os dados vinculados ao CEP informado.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/consultar/cep/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o CEP do cliente, para que seja possível consultar os dados do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para a consulta de dados do CEP.

```

1 {
2   "PARAM_NUMERO_CEP": "29145120"
3 }

```

Vale ressaltar que o número de CEP pode ser enviado **com ou sem máscara**. A mesma será tratada no WebService.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": {
3     "BAIRRO": "SANTA BARBARA",
4     "LOGRADOURO": "FRANCISCA DE PAULA BONADIMAN",
5     "UF": "ES",
6     "CIDADE": "CARIACICA"
7   },
8   "RETORNO": 0
9 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.4 - Consultar Escolaridade

A funcionalidade “consultar Escolaridade” tem a finalidade, como o próprio nome já diz, de devolver os diferentes tipos de escolaridades disponíveis.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/consultar/escolaridade/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": [
3     [
4       1,
5       "Ensino Fundamental"
6     ],
7     [
8       2,
9       "Ensino Médio"
10    ],
11    [
12      3,
13      "Ensino Superior"
14    ]
15  ],
16  "RETORNO": 0
17 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.5 - Consultar Estado Civil

A funcionalidade “consultar Escolaridade” tem a finalidade, como o próprio nome já diz, de devolver os diferentes tipos de escolaridades disponíveis.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/consultar/estado/civil/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": [
3     [
4       "S",
5       "Solteiro"
6     ],
7     [
8       "C",
9       "Casado"
10    ],
11    [
12      "V",
13      "Viúvo"
14    ],
15    [
16      "D",
17      "Divorciado"
18    ]
19  ],
20  "RETORNO": 0
21 }

```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.6 - Cadastrar

A funcionalidade “cadastrar” tem a finalidade, como o próprio nome já diz, de cadastrar o cliente e com isso fazer o seu cartão.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/cadastrar/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o número de celular, o número de CPF, o nome do cliente, dentre outros, para que seja possível realizar o cadastro do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

```

1 {
2   "PARAM_NUMERO_CELULAR": "27999437851",
3   "PARAM_NUMERO_TELEFONE": "2733362108",
4   "PARAM_NUMERO_CPF": "13805236727",
5   "PARAM_NOME": "RODRIGO TOSTA TRANCOSO",
6   "PARAM_DATA_NASCIMENTO": "19/09/1992",
7   "PARAM_ESCOLARIDADE": 3,
8   "PARAM_ESTADO_CIVIL": "S",
9   "PARAM_SEXO": "M",
10  "PARAM_DADOS_ALTERADOS": "false",
11  "PARAM_RENDA_MENSAL": 1.000,
12  "PARAM_NUMERO_CEP": "29145120",
13  "PARAM_UF": "ES",
14  "PARAM_CIDADE": "CARIACICA",
15  "PARAM_BAIRRO": "SANTA BARBARA",
16  "PARAM_LOGRADOURO": "FRANCISCA DE PAULA BONADIMAN",
17  "PARAM_NUMERO": "55",
18  "PARAM_COMPLEMENTO": "CASA"
19 }

```

Quando o cadastro foi bem sucedido, o WebService retorna um JSON de acordo com o tipo de proposta, onde a mesma pode ser pré-pago ou pós-pago, ambas as estruturas estão demonstradas abaixo.

2.6.1 - Pré Pago

```

1 {
2   "RETORNO": 0,
3   "DADOS": {
4     "TIPO": 1,
5     "MENSAGEM": "Para início de relacionamento você receberá um
cartão pré-pago no endereço de correspondência informado. "
6   }
7 }

```

2.6.2 - Pós Pago

```

1 {
2   "RETORNO": 0,
3   "DADOS": {
4     "TIPO": 0,
5     "CODIGO": 123456,
6     "LIMITE_MIN": 80.0,
7     "LIMITE_MAX": 4000.0
8   }
9 }

```

No caso de falha no cadastro, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.7 – Enviar Documentos

A funcionalidade “enviar documentos” tem a finalidade, de receber as imagens do cliente e dos documentos do cliente, para anexa-las à proposta.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/enviar/documentos/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o código da proposta, o as imagens, para que seja possível anexar às mesmas. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

???

Quando a configuração foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

???

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO":1,
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."
4 }

```

2.8 - Consultar Datas de Vencimento

A funcionalidade “consultar datas de vencimento” tem a finalidade, como o próprio nome já diz, de devolver as datas de vencimento disponíveis no sistema.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/consultar/vencimentos/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade não requer nenhum parâmetro para ser consumida.

Quando a consulta foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "DADOS": [
3     1,
4     5,
5     10,
6     11,
7     12,
8     15,
9     20,
10    25
11  ],
12  "RETORNO": 0
13 }
```

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```

1 {
2   "RETORNO": 1,
3   "MENSAGEM": "Erro ocorrido durante o processo da funcionalidade."
4 }
```

2.9 - Configurar Limite Crédito

A funcionalidade “configurar limite crédito” tem a finalidade, como o próprio nome já diz, de configurar o limite de crédito para uma proposta.

O endereço de acesso a esta funcionalidade é:

@ <http://endereco.webservice/MastercardREST/nao/cliente/configurar/limite/>

Obs.: Onde está escrito “endereco.webservice” deve ser substituído pelo endereço onde o WebService se encontra. Este endereço está configurado no apêndice A.

Esta funcionalidade espera receber um JSON contendo o código da proposta, o limite de crédito escolhido e o dia de vencimento escolhido, para que seja possível realizar o cadastro do mesmo. Caso o JSON não contenha algum parâmetro obrigatório, o WebService irá retornar uma mensagem dizendo qual parâmetro não foi informado.

Abaixo temos a estrutura do JSON necessária para o cadastro.

???

Quando a configuração foi bem sucedida, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

???

No caso de falha na consulta, o WebService retorna um JSON contendo a estrutura demonstrada abaixo.

```
1 {  
2   "RETORNO":1,  
3   "MENSAGEM":"Erro ocorrido durante o processo da funcionalidade."  
4 }
```

Apêndice A

Endereço do Servidor: 177.125.181.172.