**AA.Net Call Center API**

**API Documentation**

**Advance America**

**Version 0.1**

Last Saved: 2/13/2015 5:28 PM

By: Lawrence Casey

Advance America Call Center Services API

Contents

[Revision History 3](#_Toc411612111)

[Call Center Services API 4](#_Toc411612112)

[API URL 4](#_Toc411612113)

[XML Structures 4](#_Toc411612114)

[Pre-Application XML 4](#_Toc411612115)

[Search Request XML 5](#_Toc411612116)

[Search Response XML 6](#_Toc411612117)

[PreApp Save Request 8](#_Toc411612118)

[PreApp Save Response 8](#_Toc411612119)

[PreApp Get Request 8](#_Toc411612120)

[PreApp Get Response 8](#_Toc411612121)

[API Functions 9](#_Toc411612122)

[Search (HTTP POST Search) 9](#_Toc411612123)

[Get Pre-Application (HTTP GET PreApp) 10](#_Toc411612124)

[Save Pre-Application (HTTP POST PreApp) 10](#_Toc411612125)

Revision History

All revisions made to this document are listed below in chronological order.

| **Date** | **Version** | **Modified By** | **Section and Description** |
| --- | --- | --- | --- |
| 2/13/2015 | 0.0 | Larry Casey | Preliminary Draft & Team Review (phone meeting with SC. Rob, Joe, Bob, Chris) |
| 2/13/2015 | 1.0 | Larry Casey | Updated based on meeting feedback. API Complete Version 1 |
| 2/13/2015 | 1.1 | Larry Casey | Added the search sequence / rules |
|  |  |  |  |

# Call Center Services API

This API provides the following features:

1. Save a ‘Pre-Application’ that hasn’t been submitted to LoanGuard and retrieve it later.
2. Customer Search – search for a customer record in the AANET web site, CNU file, EEE, possibly Kickstand, as well as previously saved pre-applications.
3. Retrieve/create a pre-application from a search result – once a customer record is located with the search function, this will return a pre-application object – prefilled with the information from the selected customer search result.

## API URL

The API is exposed via a base URL for a given environment (i.e DEV / QA / Production) appended with “/services”.

For Example, The base URL may be: http://dev.AdvanceAmerica.Net, and the API is located at:

<http://dev.advanceamerica.net/services>.

The API provides the functionality described above via RESTful services.

Note that these services are only available within the AA network and not exposed to the external internet.

The data passed to and from the API is serialized as XML. Each API call has a request and response XML structure described below.

# XML Structures

This section contains the XML structures used when communicating with the API.

## Pre-Application XML

<?xml version="1.0" encoding="utf-16"?>

<WebApplication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Accept\_SMS\_Policy>true</Accept\_SMS\_Policy>

<AccountID xsi:nil="true" />

<Address>

<AddressID xsi:nil="true" />

<AddressLine1>102 Bilkenwoord Lane</AddressLine1>

<CityTown>Pillsgrove</CityTown>

<StateProvinceID>KS</StateProvinceID>

</Address>

<ApplicationDate xsi:nil="true" />

<ApplicationType>New</ApplicationType>

<BankABANumber>000000000</BankABANumber>

<BankAccountNumber>0000000000</BankAccountNumber>

<BankID xsi:nil="true" />

<BankName>Test Bank</BankName>

<CustomerIdentityID xsi:nil="true" />

<DLNumber>12312</DLNumber>

<DLStateProvinceID>KS</DLStateProvinceID>

<DLXpr>2016-01-01T00:00:00</DLXpr>

<DOB>1954-12-10T00:00:00</DOB>

<EmailAddress>email@mail.com</EmailAddress>

<EmailID xsi:nil="true" />

<EmployerID xsi:nil="true" />

<EmployerTypeID xsi:nil="true" />

<EmploymentDate xsi:nil="true" />

<ExcessiveLoans>false</ExcessiveLoans>

<FirstName>Joe</FirstName>

<FirstPayDate>2015-02-23T00:00:00</FirstPayDate>

<GrossPayPerCheck xsi:nil="true" />

<GrossPayPerMonth xsi:nil="true" />

<IsMilitary>false</IsMilitary>

<LastName>Shmo</LastName>

<LoanID xsi:nil="true" />

<MonthsAtPrevAddr xsi:nil="true" />

<NetPayPerMonth xsi:nil="true" />

<PayFreqID>0</PayFreqID>

<PayFrequencyTypeCode>Weekly</PayFrequencyTypeCode>

<PayID xsi:nil="true" />

<PayPerCheck>1000</PayPerCheck>

<PayrollBankruptcy>false</PayrollBankruptcy>

<PayrollTypeID>Direct Deposit</PayrollTypeID>

<PhoneList>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>2144322432</PhoneNumber>

<PhoneTypeCode>Home</PhoneTypeCode>

</WebPhone>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>3144322432</PhoneNumber>

<PhoneTypeCode>Mobile</PhoneTypeCode>

</WebPhone>

</PhoneList>

<ProductKitID xsi:nil="true" />

<RequestedAmount xsi:nil="true" />

<SSN>123121234</SSN>

<SecondPayDate xsi:nil="true" />

<WebApplicationID xsi:nil="true" />

<YearsAtPrevAddr xsi:nil="true" />

</WebApplication>

## Search Request XML

This is the XML structure for a search request. Searches will use SSN, then Email. The use of the DOB field is TBD.

To perform a search issue an HTTP POST to the URL <base URL>/Search

<?xml version="1.0" encoding="utf-16"?>

<CustSearchRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<SSN>123121234</SSN>

<Email>search@email.com</Email>

<DOB>1923-03-05T00:00:00</DOB>

</CustSearchRequest>

## Search Response XML

The following XML will be returned from a search request.

There is a Status section that indicates if any errors occurred, and the search results section containing the results.

The results section contains a list of CustSearchResultItem nodes.

These items contain information about the record that was found including the name, SSN, email, DOB etc. The source of the record (i.e. CNU, AA, Saved pre-app file etc. ) is in the CustType field.

Based on the search rules, the ‘selected’ app (Selected node = true) will be contain a populated “PreApp” node which will contain the complete pre-application for use when pre-populating the loan application. Other results with ‘Selected = false’ will not have the PreApp node populated.

<?xml version="1.0" encoding="utf-16"?>

<CustSearchResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Status>

<Success>true</Success>

<ErrorMessage />

</Status>

<SearchResults>

<CustSearchResultItem>

<CustID>1</CustID>

<CustType>CNU</CustType>

<CustStatus>PIF</CustStatus>

<SSN>123133243</SSN>

<FirstName>first</FirstName>

<LastName>last</LastName>

<DOB>1923-10-01T00:00:00</DOB>

<Email>email@email.com</Email>

<Selected>true</Selected>

<PreApp>

<Accept\_SMS\_Policy>true</Accept\_SMS\_Policy>

<AccountID xsi:nil="true" />

<Address>

<AddressID xsi:nil="true" />

<AddressLine1>102 Bilkenwoord Lane</AddressLine1>

<CityTown>Pillsgrove</CityTown>

<StateProvinceID>KS</StateProvinceID>

</Address>

<ApplicationDate xsi:nil="true" />

<ApplicationType>New</ApplicationType>

<BankABANumber>000000000</BankABANumber>

<BankAccountNumber>0000000000</BankAccountNumber>

<BankID xsi:nil="true" />

<BankName>Test Bank</BankName>

<CustomerIdentityID xsi:nil="true" />

<DLNumber>12312</DLNumber>

<DLStateProvinceID>KS</DLStateProvinceID>

<DLXpr>2016-01-01T00:00:00</DLXpr>

<DOB>1954-12-10T00:00:00</DOB>

<EmailAddress>email@mail.com</EmailAddress>

<EmailID xsi:nil="true" />

<EmployerID xsi:nil="true" />

<EmployerTypeID xsi:nil="true" />

<EmploymentDate xsi:nil="true" />

<ExcessiveLoans>false</ExcessiveLoans>

<FirstName>Joe</FirstName>

<FirstPayDate>2015-02-23T00:00:00</FirstPayDate>

<GrossPayPerCheck xsi:nil="true" />

<GrossPayPerMonth xsi:nil="true" />

<IsMilitary>false</IsMilitary>

<LastName>Shmo</LastName>

<LoanID xsi:nil="true" />

<MonthsAtPrevAddr xsi:nil="true" />

<NetPayPerMonth xsi:nil="true" />

<PayFreqID>0</PayFreqID>

<PayFrequencyTypeCode>Weekly</PayFrequencyTypeCode>

<PayID xsi:nil="true" />

<PayPerCheck>1000</PayPerCheck>

<PayrollBankruptcy>false</PayrollBankruptcy>

<PayrollTypeID>Direct Deposit</PayrollTypeID>

<PhoneList>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>2144322432</PhoneNumber>

<PhoneTypeCode>Home</PhoneTypeCode>

</WebPhone>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>3144322432</PhoneNumber>

<PhoneTypeCode>Mobile</PhoneTypeCode>

</WebPhone>

</PhoneList>

<ProductKitID xsi:nil="true" />

<RequestedAmount xsi:nil="true" />

<SSN>123121234</SSN>

<SecondPayDate xsi:nil="true" />

<WebApplicationID xsi:nil="true" />

<YearsAtPrevAddr xsi:nil="true" />

</PreApp>

</CustSearchResultItem>

<CustSearchResultItem>

<CustID>1</CustID>

<CustType>EEE</CustType>

<CustStatus>DEFAULT</CustStatus>

<SSN>123133243</SSN>

<FirstName>first</FirstName>

<LastName>last</LastName>

<DOB>1923-10-01T00:00:00</DOB>

<Email>email@email.com</Email>

<Selected>false</Selected>

</CustSearchResultItem>

</SearchResults>

</CustSearchResponse>

## PreApp Save Request

The PreApp save request uses the Pre-Application XML structure described above. Posting this structure to the PreApp URL will save the application.

## PreApp Save Response

The PreApp save response contains a Status node and the CustID under which the app was saved.

<?xml version="1.0" encoding="utf-16"?>

<CustSaveResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Status>

<Success>true</Success>

<ErrorMessage />

</Status>

<CustID>cnu\_123</CustID>

</CustSaveResponse>

## PreApp Get Request

The PreApp HTTP GET request has no XML structure. The ID of the PreApp to retrieve is passed in the query string. See details below.

## PreApp Get Response

<?xml version="1.0" encoding="utf-16"?>

<CustGetResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Status>

<Success>true</Success>

<ErrorMessage />

</Status>

<WebApplication>

<Accept\_SMS\_Policy>true</Accept\_SMS\_Policy>

<AccountID xsi:nil="true" />

<Address>

<AddressID xsi:nil="true" />

<AddressLine1>102 Bilkenwoord Lane</AddressLine1>

<CityTown>Pillsgrove</CityTown>

<StateProvinceID>KS</StateProvinceID>

</Address>

<ApplicationDate xsi:nil="true" />

<ApplicationType>New</ApplicationType>

<BankABANumber>000000000</BankABANumber>

<BankAccountNumber>0000000000</BankAccountNumber>

<BankID xsi:nil="true" />

<BankName>Test Bank</BankName>

<CustomerIdentityID xsi:nil="true" />

<DLNumber>12312</DLNumber>

<DLStateProvinceID>KS</DLStateProvinceID>

<DLXpr>2016-01-01T00:00:00</DLXpr>

<DOB>1954-12-10T00:00:00</DOB>

<EmailAddress>email@mail.com</EmailAddress>

<EmailID xsi:nil="true" />

<EmployerID xsi:nil="true" />

<EmployerTypeID xsi:nil="true" />

<EmploymentDate xsi:nil="true" />

<ExcessiveLoans>false</ExcessiveLoans>

<FirstName>Joe</FirstName>

<FirstPayDate>2015-02-23T00:00:00</FirstPayDate>

<GrossPayPerCheck xsi:nil="true" />

<GrossPayPerMonth xsi:nil="true" />

<IsMilitary>false</IsMilitary>

<LastName>Shmo</LastName>

<LoanID xsi:nil="true" />

<MonthsAtPrevAddr xsi:nil="true" />

<NetPayPerMonth xsi:nil="true" />

<PayFreqID>0</PayFreqID>

<PayFrequencyTypeCode>Weekly</PayFrequencyTypeCode>

<PayID xsi:nil="true" />

<PayPerCheck>1000</PayPerCheck>

<PayrollBankruptcy>false</PayrollBankruptcy>

<PayrollTypeID>Direct Deposit</PayrollTypeID>

<PhoneList>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>2144322432</PhoneNumber>

<PhoneTypeCode>Home</PhoneTypeCode>

</WebPhone>

<WebPhone>

<CountryCode>1</CountryCode>

<PhoneID xsi:nil="true" />

<PhoneNumber>3144322432</PhoneNumber>

<PhoneTypeCode>Mobile</PhoneTypeCode>

</WebPhone>

</PhoneList>

<ProductKitID xsi:nil="true" />

<RequestedAmount xsi:nil="true" />

<SSN>123121234</SSN>

<SecondPayDate xsi:nil="true" />

<WebApplicationID xsi:nil="true" />

<YearsAtPrevAddr xsi:nil="true" />

</WebApplication>

</CustGetResponse>

# API Functions

## Search (HTTP POST Search)

To search for a given customer post the search request XML described above to the base URL + “/Search”.

Ex: <http://api.advanceamerica.net/Services/Search>

The server will return the search response XML described above.

The search returns customer information as follows:

1. If a customer is found to have an account in the AA.NET web site, their most current information is returned in the WebApplication node of the response XML
2. If a record is found that was previously saved in the Pre-App file, this record will be used if the information is more current then the data found in the AA.NET web site (or if no record is found in AA.NET).
3. If the above searches don’t locate the customer, the CNU file will be checked for a match. If no record is returned, the SearchResults collection will be empty.

## Get Pre-Application (HTTP GET PreApp)

To retrieve a specific customer pre-application, perform an HTTP GET on the API URL + /PreApp/<CustID>

Use the CustID string that was returned in the search results XML CustSearchResultItem.CustID node.

Ex: <http://api.advanceamerica.net/Services/PreApp/cnu_1234>

The above URL will return the pre-application XML for customer cnu\_1234.

## Save Pre-Application (HTTP POST PreApp)

To save a pre-application, perform an HTTP POST on the API URL + /PreApp

Post XML structured like the Pre-Application described above.

The API will return a CustSaveResponse XML structure.