name.com Name.com API V2

Added by David Mc Breen, last edited by Art Enke on Jul 15, 2011

General Actions

Action	HTTP Request Type and URL
Login	POST /api/login
Logout	GET /api/logout

Hello	GET /api/hello

Login

Request: POST /api/login

```
{
  "username" : "<username>",
  "api_token" : "<api_token>"
}
```

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  },
  "session_token" : "5aa8f811f83763414c55a62a341ff808b4c10631"
}
```

Logout

Request: GET /api/logout

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  }
}
```

Hello

Request: GET /api/hello

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  },
  "service"     : "Name.com API Test Server",
  "server_date" : "2010-06-27 04:32:58",
  "version"     : "2.0",
  "language"    : "en",
  "client_ip"   : "127.0.0.1"
}
```

Account Actions

Action	HTTP Request Type and URL
Retrieve Account	GET /api/account/get

Retrieve Account

Request: GET /api/account/get

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
  "username" : "jdoe",
  "create_date" : "2010-05-26 13:23:02",
  "domain_count" : "11",
  "account_credit" : "100.00",
  "contacts" : [
    {
      "type" : ["registrant","administrative","billing","technical"],
      "first_name" : "John",
      "last_name" : "Doe",
      "organization" : "Name.com",
      "address_1" : "125 Rampart Way",
      "address_2" : "",
      "city" : "Denver",
      "state" : "CO",
      "zip" : "80230",
      "country" : "US",
      "phone" : "+1.3035555555",
      "fax" : "",
      "email" : "john@example.com"
    }
  ]
}
```

Order Actions

Action	HTTP Request Type and URL
Process Order	POST /api/order

Product Descriptions

Below is a list of available products along with a description of required and optional parameters for each. These products can be purchased using the "Process Order" command described below.

Domain Create (Registration)

Parameters:

- **order_type**: domain/create
- **domain_name**: The domain name being registered.
- **period**: The length of time for which the domain is being registered (in years).
- **nameservers** (optional): The nameservers to be applied to the domain (will default to default nameservers if not provided).
- **contact** (optional): The contacts to be applied to the domain (will default to default account contacts if not provided).
- **whois_privacy** (optional): Can be set to 1 to purchase whois privacy along with a registration (this can be used instead of individually listing a whois privacy creation - do not use both). The whois privacy registration/renewal will be itemized in the response.

Domain Renew

Parameters:

- **order_type**: domain/renew
- **domain_name**: The domain name being renewed.
- **period**: The length of time for which the domain is being renewed (in years).
- **whois_privacy** (optional): Can be set to 1 to purchase or renew whois privacy along with a renewal (this can be used instead of individually listing a whois privacy creation or renewal - do not use both). The whois privacy registration/renewal will be itemized in the response.

Whois Privacy Create

> ℹ This can alternately be included in an order using the whois_privacy flag with a domain registration or renewal. It is not necessary to specify the whois_privacy flag AND specify the stand alone whois_privacy item for an individual domain. One will fail if both are specified.

Parameters:

- **order_type**: whois_privacy/create
- **domain_name**: The domain name for which whois privacy is being created.
- **period**: The length of time for which whois privacy is being created (in years).

Whois Privacy Renew

> This can alternately be included in an order using the whois_privacy flag with a registration or renewal. It is not necessary to use the whois_privacy flag AND the stand alone whois_privacy item for an individual domain. One will fail if both are specified or it can cause the whois_privacy to be renewed multiple times.

Parameters:

- **order_type**: whois_privacy/renew
- **domain_name**: The domain name for which whois privacy is being renewed.
- **period**: The length of time for which whois privacy is being renewed (in years).

Backorder

Parameters:

- **order_type**: backorder/create
- **domain_name**: The domain name being backordered.

Process Order

The order command can be used to purchase all products available through the API. Multiple products can be purchased at one time using this command. Each product is listed as an individual entry inside the items array parameter. Each item must include all non-optional fields specified in the "Product Descriptions" section above. The items array must contain 1 or more products. The sample usage below includes multiple items as an example.

Request: POST /api/order

```
{
  "items":
  [
    {
      "order_type" : "domain/create",
      "domain_name" : "testdomain1.com",
      "nameservers" : ["ns1.name.com","ns2.name.com","ns3.name.com","ns4.name.com"],
      "contacts" : [
        {
          "type" : ["registrant","administrative","billing","technical"],
          "first_name" : "John",
          "last_name" : "Doe",
          "organization" : "Name.com",
          "address_1" : "125 Rampart Way",
          "address_2" : "",
          "city" : "Denver",
          "state" : "CO",
          "zip" : "80230",
          "country" : "US",
          "phone" : "+1.3035555555",
          "fax" : "",
          "email" : "john@example.com"
        }
      ],
      "period" : 1
    },
    {
      "order_type" : "domain/create",
      "domain_name" : "testdomain2.org",
      "period" : 1,
      "whois_privacy" : 1
    },
    {
      "order_type" : "domain/renew",
      "domain_name" : "testdomain3.org",
      "period" : 1,
      "whois_privacy" : 1
    },
    {
      "order_type" : "whois_privacy/create",
      "domain_name" : "testdomain4.com",
      "period" : 1
    },
    {
      "order_type" : "whois_privacy/renew",
      "domain_name" : "testdomain5.org",
      "period" : 1
    },
    {
      "order_type" : "backorder/create",
      "domain_name" : "testbackorder1.org"
    }
  ]
}
```

Response:

```
{
  "result":
  {
    "code":100,
    "message":"Command Successful"
  },
  "results":
  [
    {
      "success":false,
      "order_type":"domain/create",
      "domain_name":"testdomain1.com",
      "price":"9.99",
      "duration":"1",
      "interval":"year",
      "information":["Parameter Value Error - Domain Name Is Not Available"]
    },
    {
      "success":true,
      "order_type":"domain/create",
      "domain_name":"testdomain2.org",
      "price":"7.49",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"domain/renewal",
      "domain_name":"testdomain3.org",
      "price":"8.99",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"whois_privacy/create",
      "domain_name":"testdomain4.com",
      "price":"1.99",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"whois_privacy/renewal",
      "domain_name":"testdomain5.org",
      "price":"1.99",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"whois_privacy/create",
      "domain_name":"testdomain2.org",
      "price":"1.99",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"whois_privacy/renewal",
      "domain_name":"testdomain3.org",
      "price":"1.99",
      "duration":"1",
      "interval":"year"
    },
    {
      "success":true,
      "order_type":"backorder/create",
      "domain_name":"testbackorder1.org",
      "price":"49.95",
      "period":"1",
      "interval":"year"
    }
  ]
}
```

Domain Actions

Action	HTTP Request Type and URL
Create Domain	POST /api/domain/create
Retrieve Domain	GET /api/domain/get/domainName
List Domains	GET /api/domain/list
Lock Domain	POST /api/domain/lock/domainName
Unlock Domain	POST /api/domain/unlock/domainName
Update Nameservers	POST /api/domain/update_nameservers/domainName
Update Contacts	POST /api/domain/update_contacts/domainName
Check Domain Availability	POST /api/domain/check

Create Domain

ℹ The order command explained earlier in this document can be used in place of this command. The order command is also capable of registering multiple domains at one time along with all other products available through the API (domain renewal, private whois, etc).

ℹ The following TLDs have special requirements for domain creation: ASIA, CA, DE, EU, JOBS, NL, PRO, US, TEL, and TRAVEL. For creating domains with these TLDs, please refer to the following page: API Domain Create Extra Requirements.

Request: POST /api/domain/create

```
{
  "domain_name" : "<domain_name>",
  "period"      : <period>,
  "nameservers" : ["<nameserver_1>", "<nameserver_2>", "<nameserver_3>", "
<nameserver_4>"],
  "contacts"    : [
    {
      "type"         : ["registrant", "administrative", "billing", "technical"],
      "first_name"   : "<first_name>",
      "last_name"    : "<last_name>",
      "organization" : "<organization>",
      "address_1"    : "<address_1>",
      "address_2"    : "<address_2>",
      "city"         : "<city>",
      "state"        : "<state>",
      "zip"          : "<zip>",
      "country"      : "<country>",
      "phone"        : "<phone>",
      "fax"          : "<fax>",
      "email"        : "<email>"
    }
  ]
}
```

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  }
}
```

Retrieve Domain

Request: GET /api/domain/get/domainName

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  },
  "domain_name" : "testdomain1.com",
  "create_date" : "2010-06-21 20:54:55",
  "expire_date" : "2011-06-21 20:54:55",
  "locked"      : false,
  "nameservers" : ["ns1.name.com", "ns2.name.com"],
  "username"    : "testaccount",
  "contacts"    : [
    {
      "type"         : ["registrant","administrative","technical","billing"],
      "first_name"   : "John",
      "last_name"    : "Doe",
      "organization" : "Name.com",
      "address_1"    : "200 East Colfax",
      "address_2"    : "Suite 100",
      "city"         : "Denver",
      "state"        : "CO",
      "zip"          : "80203",
      "country"      : "US",
      "phone"        : "+1.3035555555",
      "fax"          : "+1.3035555556",
      "email"        : "noreply@name.com"
    }
  ]
}
```

List Domains

Request: GET /api/domain/list

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  },
  "domains" : {
    "testdomain1.com": {
      "tld" : "org",
      "create_date" : "2010-12-06 23:27:50",
      "expire_date" : "2012-12-06 23:27:50",
      "whois_privacy" : {
        "enabled" : true,
        "expire_date" : "2011-12-06 16:28:14"
      },
      "addons": {
        "domain/renew" : {
          "price" : "8.99"
        },
        "whois_privacy/renew" : {
          "price" : "1.99"
        }
      }
    },
    "testdomain2.com" : {
      "tld" : "com",
      "create_date" : "2010-12-07 22:54:46",
      "expire_date" : "2011-12-07 22:54:46",
      "addons" : {
        "whois_privacy" : {
          "price" : "1.99"
        },
        "domain/renew" : {
          "price" : "9.99"
        }
      }
    }
  }
}
```

Lock Domain

Request: POST /api/domain/lock/domainName

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  }
}
```

Unlock Domain

Request: POST /api/domain/unlock/domainName

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  }
}
```

Update Nameservers

Request: POST /api/domain/update_nameservers/domainName

```
{
  "nameservers" : ["<nameserver_1>","<nameserver_2>"]}
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  }
}
```

Update Contacts

Request: POST /api/domain/update_contacts/domainName

```
{
  "contacts" : [
    {
      "type" : ["registrant","administrative"],
      "first_name" : "John",
      "last_name" : "Doe",
      "organization" : "Name.com",
      "address_1" : "125 Rampart Way",
      "address_2" : "Suite 300",
      "city":"Denver",
      "state":"CO",
      "zip":"80230",
      "country":"US",
      "phone":"+1.3035555555",
      "fax":"+1.3035555556",
      "email":"john@example.net"
    }
  ]
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  }
}
```

Check Domain Availability

Request: POST /api/domain/check

```
{
  "keyword" : "largecat",
  "tlds" : ["com","net","org","info","us","biz","tel"],
  "services" : ["availability","suggested"]
}
```

Response:

```
{
  "result" : {
    "code"    : 100,
    "message" : "Command Successful"
  },
  "domains" : {
    "testdomain1.com" : {
      "avail"     : true,
      "tld"       : "com",
      "price"     : "9.99",
      "premium"   : false,
      "backorder" : false
    },
    "testdomain1.net" : {
      "avail"     : false,
      "tld"       : "net",
      "price"     : "49.95",
      "premium"   : false,
      "backorder" : true
    }
    //...
  },
  "suggested" : {
    "largecat.net" : {
      "avail" : true,
      "tld" : "net",
      "price" : "9.99",
      "premium" : false,
      "backorder" : false
    }
    //...
  }
}
```

DNS Actions

Action	HTTP Request Type and URL
List DNS Records	GET /api/dns/list/domainName
Create DNS Record	POST /api/dns/create/domainName
Delete DNS Record	POST /api/dns/delete/domainName

List DNS Records

Request: GET /api/dns/list/domainName

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
  "records" : [
    {
      "record_id" : "32",
      "name" : "mail.testdomain1.com",
      "type" : "MX",
      "content" : "mx0.name.com",
      "ttl" : "300",
      "create_date" : "2010-09-16 09:11:02"},
      "priority" : "10"
    },
    // ...
  ]
}
```

Create DNS Record

Request: POST /api/dns/create/domainName

```
{
  "hostname" : "mail",
  "type" : "MX",
  "content" : "127.0.0.1",
  "ttl" : 300,
  "priority" : 10
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
  "record_id" : 43,
  "name" : "mail.api-domain-a2.org",
  "type" : "MX",
  "content" : "127.0.0.1",
  "ttl" : 300,
  "create_date" : "2010-09-22 07:56:28",
  "priority" : 10
}
```

Delete DNS Record

Request: POST /api/dns/delete/domainName

```
{
  "record_id" : <record_id>
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  }
}
```

Registered Nameserver Actions

Action	HTTP Request Type and URL
List Host Records	GET /api/host/list/domainName
Sync Host Records	GET /api/host/sync/domainName
Create Host Record	POST /api/host/create/domainName
Delete Host Record	POST /api/host/delete/domainName
Add IP to Host Record	POST /api/host/add_ip/domainName
Remove IP from Host Record	POST /api/host/remove_ip/domainName

List Host Records

Request: GET /api/host/list/domainName

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
  "hosts" : {
    "ns1.mydomain.com" : [
      "127.0.0.1",
      // ...
    ],
    // ...
  }
}
```

Sync Host Records

Request: GET /api/host/sync/domainName

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
}
```

Create Host Records

Request: POST /api/host/create/domainName

```
{
  "hostname" : "ns1.mydomain.com",
  "ip" : "127.0.0.1",
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
}
```

Delete Host Records

Request: POST /api/host/delete/domainName

```
{
  "hostname" : "ns1.mydomain.com",
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
}
```

Add IP to Host Records

Request: POST /api/host/add_ip/domainName

```
{
  "hostname" : "ns1.mydomain.com",
  "ip" : "127.0.0.1",
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
}
```

Remove IP from Host Records

Request: POST /api/host/remove_ip/domainName

```
{
  "hostname" : "ns1.mydomain.com",
  "ip" : "127.0.0.1",
}
```

Response:

```
{
  "result" : {
    "code" : 100,
    "message" : "Command Successful"
  },
}
```

Authentication

Most commands require authentication (with the exception of Login, Hello, and Get Token). You can either use the login command to get a Session Token and pass that with each request or pass the Username and API Token with each request.

This is done using HTTP Request headers.

Session Token

Api-Session-Token: <session_token>

Username and API Token

Api-Username: <username>
Api-Token: <api_token>

Response Codes and Messages

Code	Message
100	Command Successful
203	Required Parameter Missing
204	Parameter Value Error
211	Invalid Command URL
221	Authorization Error
240	Command Failed
250	Unexpected Error (Exception)
251	Authentication Error
260	Insufficient Funds
261	Unable to Authorize Funds