

Name.com API V2

- General Actions
 - Login
 - Logout
 - Hello
- Account Actions
 - Retrieve Account
- Domain Actions
 - Create Domain
 - Retrieve Domain
 - List Domains
 - Lock Domain
 - Unlock Domain
 - Update Nameservers
 - Update Contacts
 - Check Domain Availability
- DNS Actions
 - List DNS Records
 - Create DNS Record
 - Delete DNS Record
- Authentication
 - Session Token
 - Username and API Token
- Response Codes and Messages

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"
    }
  ]
}
```

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

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": {
      "create_date": "2010-06-21 20:54:55",
      "expire_date": "2011-06-21 20:54:55"
    },
    "testdomain2.com": {
      "create_date": "2010-06-21 21:57:31",
      "expire_date": "2013-06-21 21:57:31"
    }
  }
}
```

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"
  }
}
```

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
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