**Authenticate User**This method authenticates the user’s credentials. Basically, a bare-bones authentication request.

Endpoint: api/Authentication

Method: Post

Format: Json

Required Values: UserId  
 Credential1  
 Credential2   
 Credential3  
 Credential4  
 Provider

Success Status Code: HttpStatusCode.NoContent

Returns: N/A

Request Class: BaseAuthenticationQuery

Result Class: N/A

Json Input Example:

{"UserId":"username","Credential1":"passphrase","Credential2":"cred2","Credential3":"cred3","Credential4":"cred4","Provider":"incommontest.org"}

Json Output Example: N/A

**Get Computer Name**This method determines a valid computer name for the computer. The intention is that the server will be able to query the domain network to determine whether the suggested name is already in use, and, if it is, suggest an alternative name to use.

Endpoint: api/computername/

Method: Get

Format: Json

Required Values: name (request parameter)

Success Status Code: HttpStatusCode.Ok

Returns: Recommended computer name to use

Result Class: DataContracts.ComputerNameResult

Notes: We should probably change the request uri so it’s not so redundant

Json Input Example: N/A

Get Url Example: https://example.server.org/api/computername/IN-DENTBMOBEL1?name=IN-DENTBMOBEL1"

Json Output Example:{"$id":"1","NameToUse":"IN-DENTBMOBEL1"}

**Get Campus**Queries the server for the user’s campus. The intent of this method is not to get the user’s current, physical location, but to get his or her conceptual location in some larger organizational framework. For example, if one of our users has a campus of “Bloomington,” this method will return “Bloomington” regardless of where, physically, the user is running the engine from.

Endpoint: api/campus/

Method: Get

Format: Json

Required Values: username (as request parameter)

Success Status Code: HttpStatusCode.Ok

Returns: campus string

Request Class: N/A

Result Class: DataContracts.Campus

Notes: 1) In our implementation, calling this class assigns a campus to the user in the database.  
2) This method differs from Get location in that the latter determines the users’ physical location, whereas this method determines the user’s conceptual location in some larger organizational framework.  
3) We should probably rework this request URL so it’s not so redundant.

Json Input Example: N/A

Get Url Example:{https://example.server.org /api/campus/bmoberly?username=bmoberly}

Json Output Example:{"$id":"1","Id":37,"ShortName":"BL","LongName":"Bloomington"}

**Get Status**Returns status information (client ip address, current GMT time, server version, minimum client version)

Endpoint: api/Status

Method: Get

Format: Json

Required Values: N/A

Success Status Code: HttpStatusCode.Ok

Returns: status information

Result Class: DataContracts.StatusInfo

Json Input Example: N/A

Get Url Example: https://server.example.org/api/status

Json Output Example:

{"$id":"1","ClientIPV4Address":"156.56.177.189","GmtTime":"Wed Jun 19 21:30:23 2013","ServerVersion":"0.114.1723.0","MinClientVersion":"13.2.0.0"}

**Get Location**Queries the server for the computer’s physical location. The intent of this method is to provide a way for the client to know where, geographically, it is running. We use this because we have to support multiple campuses as well as a general “off-campus” location.

Endpoint: api/location/

Method: Get

Format: Json

Required Values: N/A

Success Status Code: HttpStatusCode.Ok

Returns: location string

Request Class: N/A

Result Class: DataContracts.LocationQueryResult

Notes: Get location differs from Set Campus in that the latter is meant to deal with where the user is, conceptually, in some larger organizational framework. This method attempts to determine where they are physically.

Json Input Example: N/A

Get Url Example: https://example.server.org/api/location/

Json Output Example:{"$id":"1","Location":"BL"}

**Identify User**This method authenticates the user’s credentials and returns additional information about the user. Use it in place of authenticate user if you need to retrieve properties and domain groups associated with the user. The intention is that the server be able to query ADS or some other identity / preferences provider and return the results.

Endpoint: api/IdentityQuery

Method: Post

Format: Json

Required Values: AuthenticationQuery.UserId  
 AuthenticationQuery.Credential1  
 AuthenticationQuery.Credential2  
 AuthenticationQuery.Credential3  
 AuthenticationQuery.Credential4  
 AuthenticationQuery.Provider

Success Status Code: HttpStatusCode.Ok

Returns: Json Serialized text containing lists of matching properties, groups, and preferences

Request Class: DataContracts.AuthenticatedIdentityQuery

Result Class: DataContracts.IdentityQueryResult

Notes: 1) Properties is a list of string properties that you want values for. The result object will include a key/value pair dictionary for each property included in the request properties list.  
2) Groups is a list of groups to match. The resulting groups list will return the paths to the matched groups. If the request contains a group path but the result does not contain a group path, the user is not associated with that group path.  
3) Preferences have not been implemented yet, but they will work like properties.

Json Input Example:

{"AuthenticationQuery":{"UserId":"username","Credential1":"passphrase","Credential2":"cred2","Credential3":"cred3","Credential4":"cred4","Provider":"incommontest.org"},"IdentityQuery":{"Properties":[],"GroupPaths":["CN=IU-UITS-MANAGED-KB-ALL,OU=Groups,OU=Managed,DC=ads,DC=iu,DC=edu","CN=BL-BUS-KSBGradStudents,OU=BL-BUS-Student Groups,OU=BL-BUS,OU=BL,DC=ads,DC=iu,DC=edu","CN=BL-SPEA-Grad-Online,OU=Student Distribution Lists,OU=User Groups,OU=BL-SPEA,OU=BL,DC=ads,DC=iu,DC=edu","CN=IN-DENT-CLASS2015,OU=USER GROUPS,OU=GROUPS,OU=IN-DENT,OU=IN,DC=ads,DC=iu,DC=edu","CN=IN-DENT-CLASS2015,OU=IN-DENT-USER GROUPS,OU=IN-DENT,OU=IN,DC=ads,DC=iu,DC=edu"],"Preferences":null}}

Json Result Example:

{"$id":"1","Properties":{"$id":"2"},"Groups":["CN=BL-BUS-KSBGRADSTUDENTS,OU=BL-BUS-STUDENT GROUPS,OU=BL-BUS,OU=BL,DC=ADS,DC=IU,DC=EDU","CN=IU-UITS-MANAGED-KB-ALL,OU=GROUPS,OU=MANAGED,DC=ADS,DC=IU,DC=EDU"],"Preferences":null}

**Register Computer**This method registers the user’s network adapters.

Endpoint: api/Registration

Method: Post

Format: Json

Required Values: AuthenticationQuery.UserId  
 AuthenticationQuery.Credential1  
 AuthenticationQuery.Credential2  
 AuthenticationQuery.Credential3  
 AuthenticationQuery.Credential4  
 AuthenticationQuery.Provider  
 Machine.MachineId  
 Session  
 WiredAddresses  
 WirelessAddresses

Success Status Code: HttpStatusCode.NoContent

Returns: N/A

Request Class: RegistrationRequest

Result Class: N/A

Notes: Mac addresses should not be delimited

Json Input Example:

{"Machine":{"Id":0,"MachineId":"5Y17BM1"},"AuthenticationQuery":{"UserId":"username","Credential1":"passphrase","Credential2":"cred2","Credential3":"cred3","Credential4":"cred4","Provider":"incommontest.org"},"WiredAddresses":[{"Id":0,"Address":"002564C98498","Description":"Broadcom NetXtreme Gigabit Ethernet #2"},{"Id":0,"Address":"0010185CC4E1","Description":"Broadcom NetXtreme Gigabit Ethernet"}],"WirelessAddresses":[{"Id":0,"Address":"0013F72D23C5","Description":"Qualcomm Atheros AR5005GS Wireless Network Adapter"}],"Session":"86fa73e1-7936-465e-839f-a4797c810349"}

Json Result Example: N/A

**Upload Log Events**This method records log events for the client.

Endpoint: api/Logging

Method: Post

Format: Json

Required values: Text  
 Session.SessionGuid  
 EventType.Name  
 Machine.MachineId

Success Status Code HttpStatusCode.Ok

Returns: Json serialized text containing the database machineId and sessionId associated with the log entry

Request Class: DataContracts.EventEntry

Result Class: DataContracts.LogInfoWrapper

Notes: SessionGuid should be the string version of a Guid / UUID  
MachineId can be any valid string  
EventType can be any valid string, but our local implementation currently recognizes only these event type strings: Debug, Error, Info, Warn, Task, Branch Start Exit, Register, Finish, Track, Monitor, and Waypoint

Json Input Example:

{"Id":0,"Text":"Log event","Timestamp":"0001-01-01T05:00:00Z","IpAddress":null,"Session":{"Id":0,"SessionGuid":"6b1fcf40-b13f-40a7-8bb7-2a89521fd964"},"EventType":{"Id":0,"Name":"DEBUG"},"Machine":{"Id":0,"MachineId":"5Y17BM1"},"User":{"Id":0,"Username":"[Unknown]","Campus":null}}

Json Result Example:

{"$id":"1","SessionId":2089,"ComputerId":1}

**Upload Mac Address Report**Uploads a mac address report to the server. This lets us get the machine’s mac addresses before we send a registration request.

Endpoint: api/MacAddressReporting/

Method: Post

Format: Json

Required values: Machine.MachineId  
 Addresses

Success Status Code: HttpStatusCode.NoContent

Returns: N/A

Request Class: DataContracts.MachineMacAddressesReport

Result Class: N/A

Notes: This method does not currently distinguish between wired and wireless addresses. We should probably update it so that parallels register computer.

Json Input Example:

{"Machine":{"Id":0,"MachineId":"5Y17BM1"},"Addresses":[{"Id":0,"Address":"002564C98498","Description":"Broadcom NetXtreme Gigabit Ethernet #2"},{"Id":0,"Address":"0010185CC4E1","Description":"Broadcom NetXtreme Gigabit Ethernet"},{"Id":0,"Address":"0013F72D23C5","Description":"Qualcomm Atheros AR5005GS Wireless Network Adapter"}]}

Json Output Example: N/A

**Upload Report**Uploads a report to the server. Reports differ from Log entries in that they can be keyed on arbitrary values. Log entries, at least in our implementation, expect a relatively static set of event type identifiers.

EndPoint: api/reporting/

Method: Post

Format: Json

Required Values: Machine.MachineId  
 Session.SessionGuid  
 Name  
 Value

Success Status Code: HttpStatusCode.NoContent

Returns: N/A

Request Class: DataContracts.ReportingEntry

Result Class: N/A

Notes: Name = report entry key  
 Value = report entry value

Json Input Example:

{"Id":0,"Machine":{"Id":0,"MachineId":"5Y17BM1"},"Name":"campus","Value":"Bloomington","TimeStamp":"0001-01-01T05:00:00Z","Session":{"Id":0,"SessionGuid":"9c71aa07-340c-4074-8280-515333ec73c2"}}

Json Output Example: N/A

**Data Classes**

public class AuthenticatedIdentityQuery

{

public BaseAuthenticationQuery AuthenticationQuery{ get; set;}

public IdentityQuery IdentityQuery{ get; set;}

}

public class BaseAuthenticationQuery

{

public string UserId{ get; set;}

public string Credential1{ get; set;}

public string Credential2{ get; set;}

public string Credential3{ get; set;}

public string Credential4{ get; set;}

public string Provider{ get; set;}

}

public class Campus

{

public long Id{ get; set;}

public string ShortName{ get; set;}

public string LongName{ get; set;}

}

public class CampusSubnet

{

public long Id{ get; set;}

public virtual Campus Campus{ get; set;}

public string BlockStart{ get; set;}

public string BlockEnd{ get; set;}

}

public class ComputerNameResult

{

public string NameToUse{ get; set;}

}

public class ContextIdentityQuery

{

public string User{ get; set;}

public IdentityQuery IdentityQuery{ get; set;}

}

public class EventEntry

{

public long Id{ get; set;}

public string Text{ get; set;}

public DateTime Timestamp{ get; set;}

public string IpAddress{ get; set;}

public virtual Session Session{ get; set;}

public virtual EventType EventType{ get; set;}

public virtual Machine Machine{ get; set;}

public virtual User User{ get; set;}

}

public class EventType

{

public long Id{ get; set;}

public string Name{ get; set;}

}

public class FileInfoWrapper

{

public string FileUrl{ get; set;}

public string FileName{ get; set;}

public string FileSha1{ get; set;}

public long FileSize{ get; set;}

public string Version{ get; set;}

public bool Vital{ get; set;}

public string MsiProductCode{ get; set;}

public string MsiUpgradeCode{ get; set;}

[System.Xml.Serialization.XmlIgnore]

public string BaseUrl{ get; set;}

}

public class IdentityQuery

{

public List<string> Properties{ get; set;}

public List<string> GroupPaths{ get; set;}

public List<string> Preferences{ get; set;}

}

public class ReportingEntry

{

public long Id{ get; set;}

public virtual Machine Machine{ get; set;}

public string Name{ get; set;}

public string Value{ get; set;}

public DateTime TimeStamp{ get; set;}

public virtual Session Session{ get; set;}

}

public class IdentityQueryResult

{

public Dictionary<string, string> Properties{ get; set;}

public List<string> Groups{ get; set;}

public Dictionary<string, string> Preferences{ get; set;}

}

public class LocationQueryResult

{

public string Location{ get; set;}

}

public class LogInfoWrapper

{

public long SessionId{ get; set;}

public long ComputerId{ get; set;}

}

public class MacAddress

{

public long Id{ get; set;}

public string Address{ get; set;}

public string Description{ get; set;}  
}

public class MacAddressRegistration

{

public long Id{ get; set;}

public virtual MachineMacAddress MachineMacAddress{ get; set;}

public virtual User User{ get; set;}

public DateTime Timestamp{ get; set;}

}

public class Machine

{

public long Id{ get; set;}

public string MachineId{ get; set;}

}

public MachineDetails(){

public Machine Machine{ get; set;}

public List<SessionText> Sessions{ get; set;}

public List<User> Users{ get; set;}

public List<EventEntry> EventEntries{ get; set;}

public List<MacAddress> MacAddresses{ get; set;}

public List<MacAddressRegistration> Registrations{ get; set;}

public List<ReportingEntry> Reports{ get; set;}

public List<string> IpAddresses{ get; set;}

}

public class MachineMacAddress

{

public long Id{ get; set;}

public virtual Machine Machine{ get; set;}

public virtual MacAddress Address{ get; set;}

}

public class MachineMacAddressesReport

{

public Machine Machine{ get; set;}

public List<MacAddress> Addresses{ get; set;}

}

public class Preference

{

public long Id{ get; set;}

public string Name{ get; set;}

}

public class RegistrationRequest

{

public Machine Machine{ get; set;}

public BaseAuthenticationQuery AuthenticationQuery{ get; set;}

public List<MacAddress> WiredAddresses{ get; set;}

public List<MacAddress> WirelessAddresses{ get; set;}

public string Session{ get; set;}

}

public class ReportingEntry

{

public long Id{ get; set;}

public virtual Machine Machine{ get; set;}

public string Name{ get; set;}

public string Value{ get; set;}

public DateTime TimeStamp{ get; set;}

public virtual Session Session{ get; set;}

}

public class Session

{

public long Id{ get; set;}

public Guid SessionGuid{ get; set;}

}

public class SessionText

{

public Session Session{ get; set;}

public string Text{ get; set;}

}

public class StatusInfo

{

public string ClientIPV4Address{ get; set;}

public string GmtTime{ get; set;}

public string ServerVersion{ get; set;}

public string MinClientVersion{ get; set;}

}

public class SupportSearchResult

{

public List<UserMachineSession> Results{ get; set;}

public List<string> SupportedQueries;

public string LastQuery{ get; set;}

public string LastQueryType{ get; set;}

public string Issue{ get; set;}

}

public class User

{

public long Id{ get; set;}

public string Username{ get; set;}

public virtual Campus Campus{ get; set;}

}

public class UserDetails

{

public User User{ get; set;}

public List<SessionText> Sessions{ get; set;}

public List<Machine> Machines{ get; set;}

public List<EventEntry> EventEntries{ get; set;}

}

public class UserMachine

{

public long Id{ get; set;}

public virtual User User{ get; set;}

public virtual Machine Machine{ get; set;}

}

public class UserMachineSession

{

public long Id{ get; set;}

public virtual UserMachine UserMachine{ get; set;}

public virtual Session Session{ get; set;}

}

public class UserPreferenceValue

{

public long Id{ get; set;}

public virtual User User{ get; set;}

public virtual Preference Preference{ get; set;}

public string Value{ get; set;}

}