Trouble Ticketing Web Service Revised API

Types:

There are three data types in the Trouble Ticketing API (TT-API); the Trouble Ticket (TT), Incident Report (IR), and Bug Report (BR).

Trouble Ticket:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>XML Type:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket ID</td>
<td>dateTime.iso8601</td>
<td>Time ticket was opened, primary key</td>
</tr>
<tr>
<td>Owner</td>
<td>string</td>
<td>Owner of the faulty product</td>
</tr>
<tr>
<td>PartNumber</td>
<td>string</td>
<td>Part # of faulty product</td>
</tr>
<tr>
<td>CloseDate</td>
<td>dateTime.iso8601</td>
<td>Time ticket is closed</td>
</tr>
<tr>
<td>Closer</td>
<td>string</td>
<td>Person who closes the ticket</td>
</tr>
<tr>
<td>Status</td>
<td>int</td>
<td>Determines status of ticket</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
<td>Description of problem</td>
</tr>
</tbody>
</table>

Logical Ticket Types:

Pending Ticket: Newly created TT  Status = -1  - newly created with little or no Incident Reports assigned to it

Trouble Ticket: Active TT  Status = 0  - complete and active Ticket

Closed Ticket: Inactive TT  Status = 1  - a Ticket for a problem that is no longer an issue
Incident Report:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket ID</td>
<td>dateTime.iso8601</td>
<td>Ticket this IR is associated with</td>
</tr>
<tr>
<td>Incident ID</td>
<td>dateTime.iso8601</td>
<td>Time this IR was generated, primary key</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
<td>Description of the problem</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>string</td>
<td>Serial # / version of the faulty product</td>
</tr>
<tr>
<td>PartNumber</td>
<td>string</td>
<td>Part # of the faulty product</td>
</tr>
<tr>
<td>Owner</td>
<td>string</td>
<td>Owner of faulty product</td>
</tr>
</tbody>
</table>

Bug Report:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>XML Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug ID</td>
<td>dateTime.iso8601</td>
<td>Time this BR was created</td>
</tr>
<tr>
<td>PartNumber</td>
<td>string</td>
<td>Part # of buggy product</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>string</td>
<td>Serial # of product bug was initially found in</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
<td>Description of bug</td>
</tr>
<tr>
<td>Workaround</td>
<td>string</td>
<td>Workaround, if any, for this bug</td>
</tr>
</tbody>
</table>

Methods:

CreateIncidentReport(Owner, ProductNumber, SerialNumber, Description)
- create a new Incident Report; Ticket ID and Incident ID are set automatically

CreateTicket(Owner, ProductNumber, Description)
- create a new Trouble Ticket; Ticket ID and Status are set automatically

OpenTicket(Ticket ID)
- opens a Pending or Closed Ticket and makes it an Active Ticket

CloseTicket(Ticket ID, Closer)
- closes an Active Ticket and makes it an Inactive Ticket
DeleteTicket(Ticket ID)
-removes an Inactive Ticket from the system
DeleteIncident(Incident ID)
-removes an Incident Report from the system that either is not associated with a Ticket, or whose Ticket has been deleted
DeleteBug(Bug ID)
-removes a Bug Report from the system
CreateBugReport(PartNumber, SerialNumber, Description, Workaround)
-creates a new Bug Report; Bug ID is set automatically
MakeBugReport(Incident ID, Workaround)
-makes a Bug Report out of the Incident Report specified
SetIncidentTicketID(Incident ID)
-set the Ticket ID that an Incident Report belongs to
GetIncidentTicketID(Incident ID)
-returns the Ticket ID that an Incident Report belongs to
GetAll()
-returns all Tickets and Reports
GetAll(Ticket ID, Ticket ID)
-returns all Tickets for a date range
GetTicket(Ticket ID)
-returns Ticket
GetBugReport(Bug ID)
-returns Bug Report
GetAllBugReports()
-returns all Bug Reports
GetAllBugReports(Bug ID, Bug ID)
-returns all Bugs for a date range
GetAllIncidentReports()
-returns all Incident Reports
GetAllIncidentReports(Incident ID, Incident ID)
-returns all Incidents for a date range
GetAllActiveTickets()
-returns all Active Tickets
GetAllActiveTickets(Ticket ID, Ticket ID)
-returns all Active Tickets for a date range
GetAllInactiveTickets()
-returns all Inactive Tickets’ Ticket IDs
GetAllInactiveTickets(Ticket ID, Ticket ID)
-returns all Inactive Tickets for a date range
GetAllBugReports()
-returns all Bug Reports
GetAllBugReports(Bug ID, Bug ID)
-returns all Bug Reports for a date range
GetAllIncidentReportsByOwner(Owner)
-returns all Incident Reports for a specific Owner
GetAllActiveTicketsByOwner(Owner)
-returns all Active Tickets for a specific Owner
GetAllInactiveTicketsByOwner(Owner)
-returns all Inactive Tickets for a specific Owner
GetAllIncidentReportsByOwner(Owner)
-returns all Incident Reports for a specific Owner
GetAllIncidentReportsByPart(PartNumber)
-returns all Incident Reports for a specific part
GetAllActiveTicketsByPart(PartNumber)
-returns all Active Tickets for a specific part
GetAllInactiveTicketsByReporter(PartNumber)
-returns all Inactive Tickets for a specific part
GetAllBugReportsByPart(PartNumber)
-returns all Bug Reports for a specific part
GetOwner(Ticket ID / Incident ID)
-returns the Owner of a Ticket or Incident
GetPart(Ticket ID / Incident ID)
-returns the PartNumber of a Ticket or Incident
GetDescription(Ticket ID / Incident ID / Bug ID)
-returns the Description of a Ticket, Incident or Bug
GetNewestIncident()
-returns the most recently logged Incident Report
GetOldestIncident()
-returns the oldest logged Incident Report
Version()
-returns current version of the Trouble Ticketing service
GetAllProductsDeployed()
-returns all Part Numbers of currently deployed products
*requires access to the Deployment service
GetAllDeployedIncidents()
-returns all Incident Reports pertaining to any deployed product
*requires access to the Deployment service
GetAllDeployedTickets()
-returns all Tickets pertaining to any deployed product
*requires access to the Deployment service
GetAllDeployedBugs()
-returns all Bugs pertaining to any deployed product
*requires access to the Deployment service
DiscontinueProduct(PartNumber)
-closes all Tickets pertaining to the specified PartNumber and
sets their Closer to be “Fubar Sales”

Load:

In order to simulate accesses to the service, a client program will be written to
first generate Incident Reports. As the database begins to be populated it will create less
Incident Reports while executing the other web service status-changing and lookup
functions more often.