



SpringAhead API Documentation V1

Version 1.3

Created On: August 16, 2011

Modified On: April 18, 2012

Author: Mark Milbourne

Copyright SpringAhead 2011

Table of Contents

- [Table of Contents](#)
- [Document Change History](#)
- [Audience](#)
- [Introduction](#)
 - [Where is it?](#)
 - [Development Sandbox](#)
- [What is REST?](#)
- [Basic Authentication](#)
- [What is XML?](#)
- [SpringAhead API](#)
 - [Resource Not Found.](#)
 - [Input Validation](#)
 - [Example validation response.](#)
- [Endpoints](#)
 - [/](#)
 - [Timecard](#)
 - [MyTimecards](#)
 - [Expenses](#)
 - [InvoiceAccount](#)
 - [List](#)
 - [Details](#)
 - [Create](#)
 - [Edit](#)
 - [Classification](#)
 - [List](#)
 - [Details](#)
 - [Create](#)
 - [Edit](#)
 - [Company](#)
 - [List](#)
 - [Details](#)
 - [Create](#)
 - [Edit](#)
 - [MyCompany / Enterprise](#)
 - [List](#)
 - [ExpenseType](#)
 - [Item](#)
 - [InvoiceMessage](#)
 - [List](#)
 - [Details](#)
 - [Create](#)
 - [Edit](#)
 - [PaymentMethod](#)
 - [PayrollItem](#)
 - [Project](#)
 - [List](#)
 - [Details](#)
 - [Create](#)
 - [Edit](#)
 - [Task](#)
 - [Term](#)
 - [TimeType](#)
 - [User](#)
 - [List](#)
 - [Details](#)
 - [CurrentUser](#)
 - [Create](#)
 - [Edit](#)
 - [Xid](#)
 - [List](#)
 - [Details](#)
 - [Edit](#)
 - [MultiEdit](#)
 - [Create](#)
 - [Delete](#)

[Search](#)

[Secondaries](#)

[Secondaries for Projects](#)

[Rates](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[Reps](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[Budgets](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[POs](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[Tasks](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[Secondaries for Users](#)

[Rates](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[Reps](#)

- [List](#)
- [Details](#)
- [Edit](#)
- [MultiEdit](#)
- [Create](#)
- [Delete](#)
- [MultiDelete](#)

[References](#)

Document Change History

Date	Version	Author	Description
August 16, 2011	V1.0	Mark Milbourne	Initial draft of the API
December 28, 2011	v1.1	Mark Milbourne	Changed Class to Classification. Account to InvoiceAccount. Message to InvoiceMessage.
March 27, 2012	v1.2	Mark Milbourne	Fixed spelling mistake on ModifiedSince timecard API. Adding Project Rates, Project Reps, Project Tasks, Project PO's Project Budget, User Rates, User Reps documentation. Updated the documentation on the User endpoint. Updated the documentation on the MyCompany and Enterprise endpoints.
April 18, 2012	v1.3	Mark Milbourne	Fixed the links to the schemas. They were missing the v1 in the url link.

Audience

This document is written with a programmer/developer in mind. Some concepts and techniques will be assumed. Also a level of technical ability will be assumed.

Introduction

The **SpringAhead API** is a REST based service, using XML to communicate information and requests. For the first version of the **SpringAhead API** we implemented a Basic Authentication mechanism (See http://en.wikipedia.org/wiki/Basic_access_authentication). This is supported by nearly every device out there, so its a good choice as the lowest common denominator for Authentication. However, Basic Authentication does suffer from easily accessing the User credentials through a man in the middle attack. Fortunately the solution to this is simple, use SSL on all requests (See http://en.wikipedia.org/wiki/Transport_Layer_Security). As such the only way to access the **SpringAhead API** is via HTTPS.

Where is it?

The **SpringAhead API** is located at; <https://api.springahead.com/v1>

Development Sandbox

We definitely do not want you to be accessing the production **SpringAhead API** while you are developing code against it. The **SpringAhead API** is fairly rich in raw functionality and its possible to do damage to your or your customers data.

So we setup a Sandbox environment, where you can “*go to town*” on the **SpringAhead API** without fear of loosing data, or breaking something. The Sandbox **SpringAhead API** is located at;

<https://apisandbox.springahead.com/v1>

We also setup our website there, so you can logon and check that your application code using the **SpringAhead API** is working, or (hopefully not) breaking it. That URI is:

<https://apisandbox.springahead.com/go>

To obtain an account on here simply send an email to:

apirequest@springahead.com

Please specify the following information:

- Company Name
- Contact Details
- Description of what you intend to do with the API

What is REST?

REST stands for Representational State Transfer. REST is the use of HTTP as a from of a RPC mechanism to a webservice. At its core there are three main components,

- The URI, or web address. eg. <https://api.springahead.com/v1/User/1>



- The transport mechanism. In our case this is XML
- A set of operations. These are the standard HTTP verbs. We support, POST, PUT, DELETE and GET.

As an example; To get the list of Users, you would make a GET request to the URI <https://api.springahead.com/v1/Users>. Then the server (after basic auth success) would respond with XML that detailed the list of users

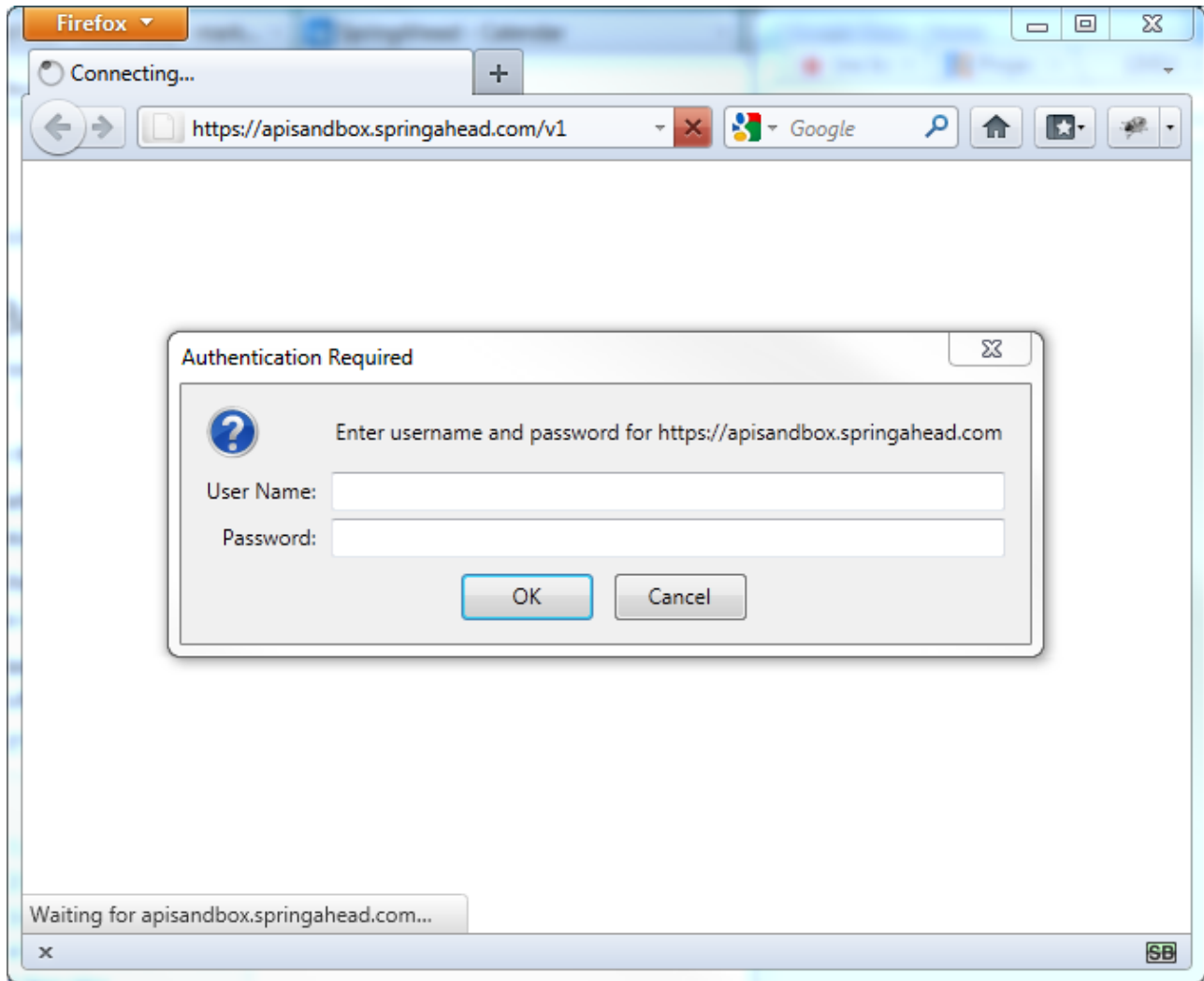
One of the advantages of REST over other communication mechanisms is that its caller and receiver agnostic. That meaning, it doesn't matter what operating system or programming language is used on each end. The protocol is well defined, simple and mature. Nearly all modern programming languages have some sort of REST library, REST features, or at worst can make HTTP requests.

For more information see http://en.wikipedia.org/wiki/Representational_State_Transfer

Basic Authentication

As we mentioned earlier the V1 **SpringAhead API** uses Basic Authentication as the mechanism to control access to the API. Nearly all browsers support this protocol, and nearly all programming languages have libraries that allow passing Basic Authentication credentials.

You can see how this works by pointing your browser to the **SpringAhead API** endpoint. <https://api.springahead.com/v1> or <https://apisandbox.springahead.com/v1>



(The image does use the apisandbox, as of writing the api.springahead.com server wasn't ready)

As you can see the Firefox browser was told that Basic Authentication is required and has prompted the user with a logon dialog (Sometimes referred to as a challenge request). This is actually making a GET request to the server. You won't be able to do the other verbs using your browser, you'll have to use something like [Fiddler2](#).

Now for those who've used SpringAhead (formerly VirtualSoftware) before know that to logon you need three things;

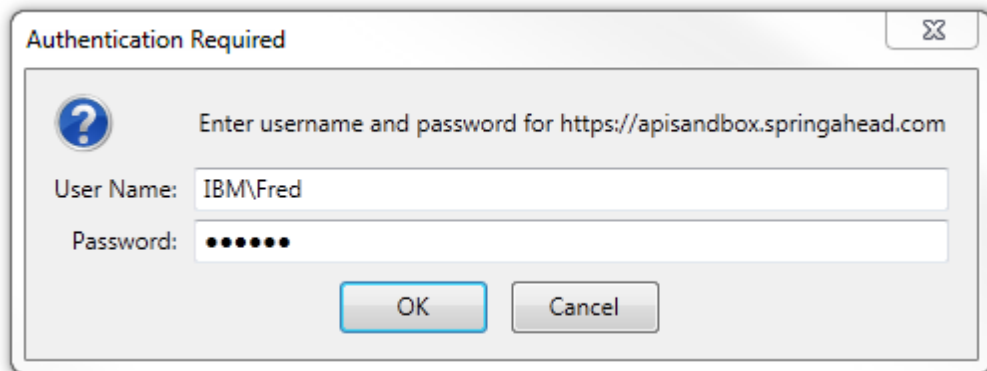
1. Company
2. Username
3. Password

And yet the logon dialog above only has *Username* and *Password*. Well, we used a trick from Microsoft. You specify the *Username* as a combination of *Company* and *Username* with a back slash (\) in between them, like this;

<CompanyName>\<UserName>

For example;

IBM\Fred
Intuit\John



When accessing any URI on the **SpringAhead API** you will need to pass Basic Authentication credentials each time. This means you'll need to cache them on your application somewhere. A point of confusion is that the browser will cache the credentials for you, so once you've entered them in on a browser you won't need to do it again. To logon as a different user you'll have to close the browser and start it again. It's a pain, but that's just how it is.

For more information on Basic Authentication; http://en.wikipedia.org/wiki/Basic_authentication

What is XML?

XML stands for Extensible markup language. For more information read here <http://en.wikipedia.org/wiki/XML>

The **SpringAhead API** uses XML to communicate all information. All responses and requests are in XML. An example response might look like;

```
<SpringAhead>
  <User>
    <Id>1</Id>
    <Active>true</Active>
    <CreatedDate>2011-03-10T10:53:25</CreatedDate>
    <CreatedUserId>1</CreatedUserId>
    <ModifiedDate>2011-08-11T17:24:23</ModifiedDate>
    <ModifiedUserId>1</ModifiedUserId>
    <LoginName>Admin</LoginName>
    <LastName>Admin</LastName>
    <CanChangePassword>true</CanChangePassword>
    <MustChangePassword>false</MustChangePassword>
    <EmployeeType>Contractor</EmployeeType>
    <Name>Admin</Name>
  </Address>
  <EnableEmailNotifications>true</EnableEmailNotifications>
  <EmailFormat>HTML</EmailFormat>
  <DateNotified xsi:nil="true"/>
  <ForceRegularPay>false</ForceRegularPay>
  <PayCycle>All</PayCycle>
  <Term>
    <Id>1</Id>
    <Active>true</Active>
    <CreatedDate>2011-03-28T13:57:43</CreatedDate>
    <CreatedUserId>1</CreatedUserId>
    <ModifiedDate>2011-03-28T13:57:43</ModifiedDate>
```




```
<ModifiedUserId>1</ModifiedUserId>
<Name>qwertyuio</Name>
<DaysDue>90</DaysDue>
</Term>
<Burden>0</Burden>
<LogoutTimeout xsi:nil="true"/>
<TimecardInfo>
  <FirstDayOfWeek xsi:nil="true"/>
  <FirstDayOfMonth xsi:nil="true"/>
</User>
</SpringAhead>
```

Note: I cut this down so as to not take up as much space in this document. The actual results are much larger.

SpringAhead API

Now we have the basics done, let's detail the API URI's and how to use them. Here are some important things about the endpoints and making requests;

- The API endpoints are not case sensitive.
- All endpoints support pluralized versions of themselves. For example /User and /Users access the same thing.
- Return codes from the API adhere to all the HTTP server codes. See <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
- When a list is returned it is in a paginated format.
- The API expects subsets of XML. So to update one field all you need to do is send us just the XML for that field. We don't require you to send the entire thing every time.
- Due to legacy issues, all Created dates and Modified dates are in PST. :(
 - Except XID, SystemMessages, UserPreferences
- Some entries will be Null in the database, these will be represented with the attribute `xsi:nil="true"`. To make something Null you can specify the same attribute. Consult the schema's for exact locations.

Resource Not Found.

If you attempt to access a resource that doesn't exist you'll get a 404 error response with the following XML;

```
<ErrorResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ResponseCode>404</ResponseCode>
  <Message>The Requested resource was not found.</Message>
</ErrorResponse>
```

All error messages from the server will fall into this format.

Input Validation

All Edit and Create endpoints will validate the input XML. This is to check the validity of the XML and to ensure that the items specified exist, and conform to the rules of our system.

Example validation response.

The entity with the most validation is the Project, so let's start with an example of how to create a new Project. The first thing we need to do is to hit the endpoint <https://api.springahead.com/v1/project> with a PUT verb, containing the following XML.

```
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Project>
    <Active>true</Active>
    <Name>aaa</Name>
  </Project>
</SpringAhead>
```

This will attempt to create a Project with the Name of "aaa" and make it Active. Notice that we don't have to send every XML element, just the ones we want to set. Everything else will get default values.

This returns with the response;

```
<?xml version="1.0" encoding="utf-8"?>
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
  <ValidationMessage>
    <Name>0:Company</Name>
    <Message>You must specify a Company for the Project.</Message>
  </ValidationMessage>
  <ValidationMessage>
    <Name>0:DateStart</Name>
    <Message>Please select a valid Date Start.</Message>
    <Message>The Date Start is out of range, please select a date before 1900.</Message>
  </ValidationMessage>
</SpringAhead>
```

and an return code of 400. So what happened here is a project has certain required fields and these requirements were not met. In this case the server is telling us that we need to set a Company, and a DateStart. So lets do that;

```
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
  <Project>
    <Active>true</Active>
    <Company>
      <Id>18</Id>
    </Company>
    <Name>aaa</Name>
    <DateStart>2011-08-02T00:00:00</DateStart>
  </Project>
</SpringAhead>
```

Since the Company is referring to another entity we only need to specify the id of the Company. You can get the list of Companies here <http://api.springahead.com/v1/companies>. Sending this to the server returns the following;

```
<?xml version="1.0" encoding="utf-8"?>
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
  <ValidationMessage>
    <Name>0:Name</Name>
    <Message>The Name doesn't conform to the convention of &lt;company name&gt;;&lt;project
name&gt;</Message>
    <Message>This change would result in a project name that already exists in the database. Check
the company and or Name change.</Message>
  </ValidationMessage>
</SpringAhead>
```

Yet another 400 response. This time the server is complaining that the name doesn't conform to the naming convention for a Project. This is <Company Name>:<Project Name>¹. See how the Name has a number prefixed to it? That tells us the position of the element in the request. If we specified more than one Project entity in the XML then the number would reference that.And yes the count starts from 0.

So lets fix our problem.

¹This is due to the convention in Quickbooks. We unfortunately are exposing this to you. This may change in the future. Project is the only entity that has this naming convention.

```
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
  <Project>
    <Active>true</Active>
    <Company>
      <Id>18</Id>
    </Company>
    <Name>Company1: aaa</Name>
    <DateStart>2011-08-02T00:00:00</DateStart>
  </Project>
</SpringAhead>
```

Now the response is;

```
<?xml version="1.0" encoding="utf-8"?>
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Project>
    <Id>55</Id>
    <Active>true</Active>
    <CreateDate>2011-08-19T13:30:25.2868942-07:00</CreateDate>
    <CreatedUserId>1</CreatedUserId>
    <ModifiedDate>2011-08-19T13:30:25.2868942-07:00</ModifiedDate>
    <ModifiedUserId>1</ModifiedUserId>
    <Company>
      <Id>18</Id>
      <Active>true</Active>
      <CreateDate>2011-07-18T14:51:53</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-07-18T14:51:53</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <Name>Company1</Name>
      <RoleCustomer>true</RoleCustomer>
      <RoleVendor>false</RoleVendor>
      <BillCycle>All</BillCycle>
      <InvoiceMessage xsi:nil="true" />
      <AccountsPayableContact xsi:nil="true" />
      <InvoiceTerm xsi:nil="true" />
      <FirstDayOfWeek>-1</FirstDayOfWeek>
      <FirstDayOfMonth>0</FirstDayOfMonth>
    </Company>
    <Name>Company1: aaa</Name>
    <DateStart>2011-08-02T00:00:00</DateStart>
    <DateStop xsi:nil="true" />
    <ApprovalSettings>
      <ManagerApprovesTime>false</ManagerApprovesTime>
      <ManagerApprovesExpense>false</ManagerApprovesExpense>
      <ApprovalAnyTime>true</ApprovalAnyTime>
      <ApprovalAnyExpense>true</ApprovalAnyExpense>
      <ApprovalExclusiveTime>false</ApprovalExclusiveTime>
      <ApprovalExclusiveExpense>false</ApprovalExclusiveExpense>
    </ApprovalSettings>
    <DepartmentId xsi:nil="true" />
    <Class xsi:nil="true" />
    <TimeTypeSettings>
      <AllowRegular>true</AllowRegular>
      <AllowOvertime>false</AllowOvertime>
      <BillOvertime>true</BillOvertime>
      <PayOvertime>true</PayOvertime>
      <AllowDoubletime>false</AllowDoubletime>
      <BillDoubletime>true</BillDoubletime>
      <PayDoubletime>true</PayDoubletime>
      <AllowSick>false</AllowSick>
      <BillSick>false</BillSick>
      <PaySick>true</PaySick>
      <AllowVacation>false</AllowVacation>
      <BillVacation>false</BillVacation>
```

```

    <PayVacation>true</PayVacation>
    <AllowHoliday>false</AllowHoliday>
    <BillHoliday>false</BillHoliday>
    <PayHoliday>true</PayHoliday>
    <AllowNonBillable>false</AllowNonBillable>
    <PayNonBillable>true</PayNonBillable>
    <AllowCustom>false</AllowCustom>
  </TimeTypeSettings>
  <TimeHourSettings>
    <OvertimeDayHours xsi:nil="true" />
    <OvertimeWeekHours xsi:nil="true" />
    <DoubletimeDayHours xsi:nil="true" />
    <LimitDayUserHours xsi:nil="true" />
    <LimitDayHours xsi:nil="true" />
    <LimitPeriodUserHours xsi:nil="true" />
    <LimitPeriodHours xsi:nil="true" />
    <TotalHours xsi:nil="true" />
  </TimeHourSettings>
  <AccountsReceivableAccount xsi:nil="true" />
  <AccountsPayableUser xsi:nil="true" />
  <Item xsi:nil="true" />
  <JobIsCompany>InvoiceTo_ThisProject</JobIsCompany>
  <AnyUserSubmitsTime>Project_Track_Selected_Time_Users</AnyUserSubmitsTime>
  <BillRateOverride xsi:nil="true" />
  <PayRateOverride xsi:nil="true" />
  <OvertimeMultiplierBill xsi:nil="true" />
  <OvertimeMultiplierPay xsi:nil="true" />
  <UseTasks>false</UseTasks>
  <TaskMandatory>Project_Task_Optional</TaskMandatory>
  <SelectTasks>false</SelectTasks>
  <EnableBillables>false</EnableBillables>
  <EnablePayables>false</EnablePayables>
  <BillCycle>All</BillCycle>
  <BillTo xsi:nil="true" />
  <ExportTo xsi:nil="true" />
  <InvoicePerUser>false</InvoicePerUser>
  <FirstDayOfWeek xsi:nil="true" />
  <FirstDayOfMonth xsi:nil="true" />
  <TimeLateIn xsi:nil="true" />
  <TimeLateOut xsi:nil="true" />
  <AllowExpense>false</AllowExpense>
  <AnyUserSubmitsExpense>Project_Track_Selected_Expense_Users</AnyUserSubmitsExpense>
  <SelectTimeTypes>false</SelectTimeTypes>
  <EnforceOT>false</EnforceOT>
  <TimeEntry xsi:nil="true" />
  <Burden xsi:nil="true" />
</Project>
</SpringAhead>

```

Success, we just created a project. The server returned all the information about the new Project. The most important info is the ID. In this case it's 55.

Things to notice are all the `xsi:nil="true"` that indicates Null in the field. This usually means that it inherits from a higher setting. Usually from the MyCompany settings.

Endpoints

/

URI Format	/
Verb	GET
Access	All
Description	This is the base endpoint of the webservice. It will give you a list of endpoints that you have access to with this account. Along with the version and and build of the service.
Example Response	<pre> <SpringAhead> <Version>8.0.842 (8052)</Version> <Product>SpringAhead API</Product> <EndPoints> <Url>https://apisandbox.springahead.com/v1/</Url> <Url>https://apisandbox.springahead.com/v1/Account</Url> <Url>https://apisandbox.springahead.com/v1/Class</Url> <Url>https://apisandbox.springahead.com/v1/Company</Url> <Url>https://apisandbox.springahead.com/v1/CurrentUser</Url> <Url>https://apisandbox.springahead.com/v1/ExpenseType</Url> <Url>https://apisandbox.springahead.com/v1/Item</Url> <Url>https://apisandbox.springahead.com/v1/Message</Url> <Url>https://apisandbox.springahead.com/v1/MyCompany</Url> <Url>https://apisandbox.springahead.com/v1/PaymentMethod</Url> <Url>https://apisandbox.springahead.com/v1/PayrollItem</Url> <Url>https://apisandbox.springahead.com/v1/Project</Url> <Url>https://apisandbox.springahead.com/v1/Task</Url> <Url>https://apisandbox.springahead.com/v1/Term</Url> <Url>https://apisandbox.springahead.com/v1/Timecard</Url> <Url>https://apisandbox.springahead.com/v1/TimeType</Url> <Url>https://apisandbox.springahead.com/v1/User</Url> <Url>https://apisandbox.springahead.com/v1/Xid</Url> </EndPoints> </SpringAhead> </pre>
Known issues	Right now the access rights are not checked, and some URLs are missing. Its on our todo list.

Timecard

URI Format	<pre> /timecard /timecards /timecard/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecard/unapproved /timecards/unapproved /timecard/unapproved/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/unapproved/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecard/range/{FromDate}/{ToDate} /timecards/range/{FromDate}/{ToDate} /timecard/range/{FromDate}/{ToDate}/{userid} /timecards/range/{FromDate}/{ToDate}/{userid} /timecard/range/{FromDate}/{ToDate}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/range/{FromDate}/{ToDate}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecard/range/{FromDate}/{ToDate}/{userid}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/range/{FromDate}/{ToDate}/{userid}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/modifiedsince/{FromDate} /timecards/modifiedsince/{FromDate}/{UserId} /timecards/modifiedsince/{FromDate}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} /timecards/modifiedsince/{FromDate}/{UserId}/page/{pageIndex}/{pagesize}/{orderby}/{ascending} </pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	https://api.springahead.com/v1/timecard/schema
Description	<p>Get a list of timecards associated with this enterprise.</p> <p>Specify dates in {year}-{month}-{day} format. It will accept others but this is the safest to avoid localization issues.</p> <p>/timecard/{FromDate}/{ToDate} - Returns a list of timecards that fall between these dates. The dates are PST. :-(</p> <p>/timecard/modifiedsince/{FromDate} - Returns a list of timecards that have been modified since the FromDate.</p>

<p>Example Response</p>	<pre> <SpringAhead> <RecordCount>9</RecordCount> <PageIndex>1</PageIndex> <PageSize>2</PageSize> <PageCount>5</PageCount> <List> <Timecard> <Id>3</Id> <CreateDate>2011-04-08T10:24:00</CreateDate> <CreatedUserId>0</CreatedUserId> <ModifiedDate>2011-04-08T10:24:00</ModifiedDate> <ModifiedUserId>0</ModifiedUserId> <Submit>true</Submit> <SubmitDate>2011-04-08T10:24:00</SubmitDate> <SubmitUserId>0</SubmitUserId> <Row>1</Row> <UserId>0</UserId> <TimecardDate>2011-04-05T00:00:00</TimecardDate> <Projectid>0</Projectid> <Taskid>-1</Taskid> <TimeType>0</TimeType> <HoursAdjust>0</HoursAdjust> <HoursBreak>0</HoursBreak> <HoursDay>8</HoursDay> <ApprovedStatus>0</ApprovedStatus> <ApprovedUserId>0</ApprovedUserId> <ApprovedText>Error: No default approver found.</ApprovedText> <DateApproved xsi:nil="true"/> <ProcessId>-1</ProcessId> <ProcessUserId>0</ProcessUserId> <ProcessDate xsi:nil="true"/> <BillableId>-1</BillableId> <BillableUserId>0</BillableUserId> <BillableDate xsi:nil="true"/> <PayableId>-1</PayableId> <PayableUserId>0</PayableUserId> <PayableDate xsi:nil="true"/> <InvoicedDate xsi:nil="true"/> <BillRateBase>0</BillRateBase> <BillRateMult>0</BillRateMult> <PayRateBase>0</PayRateBase> <PayRateMult>0</PayRateMult> <ItemId>0</ItemId> </Timecard> <Timecard> <Id>2</Id> <CreateDate>2011-04-08T10:22:00</CreateDate> <CreatedUserId>0</CreatedUserId> <ModifiedDate>2011-04-08T10:24:00</ModifiedDate> <ModifiedUserId>0</ModifiedUserId> <Submit>false</Submit> <SubmitDate>2011-04-08T10:22:00</SubmitDate> <SubmitUserId>0</SubmitUserId> <Row>0</Row> <UserId>0</UserId> <TimecardDate>2011-04-06T00:00:00</TimecardDate> <Projectid>0</Projectid> <Taskid>-1</Taskid> <TimeType>0</TimeType> <HoursAdjust>0</HoursAdjust> <HoursBreak>0</HoursBreak> <HoursDay>4</HoursDay> <ApprovedStatus>0</ApprovedStatus> <ApprovedUserId>0</ApprovedUserId> <DateApproved xsi:nil="true"/> <ProcessId>-1</ProcessId> <ProcessUserId>0</ProcessUserId> </pre>
-------------------------	--

	<pre> <ProcessDate xsi:nil="true"/> <BillableId>-1</BillableId> <BillableUserId>0</BillableUserId> <BillableDate xsi:nil="true"/> <PayableId>-1</PayableId> <PayableUserId>0</PayableUserId> <PayableDate xsi:nil="true"/> <InvoicedDate xsi:nil="true"/> <BillRateBase>0</BillRateBase> <BillRateMult>0</BillRateMult> <PayRateBase>0</PayRateBase> <PayRateMult>0</PayRateMult> <ItemId>0</ItemId> </Timecard> </List> </SpringAhead> </pre>
Example Urls	<pre> /timecard/2011-4-1/2012-1-1 /timecard/2011-4-1/2012-1-1/page/1/2 /timecard/unapproved/page/0/10 </pre>
Known issues	Currently this is Read only.

MyTimecards

URI Format	<pre> /mytimecard /mytimecards /mytimecard/page/{pageindex}/{pagesize}/{orderby}/{ascending} /mytimecards/page/{pageindex}/{pagesize}/{orderby}/{ascending} /mytimecard/range/{FromDate}/{ToDate} /mytimecard/range/{FromDate}/{ToDate}/{userid} /mytimecard/range/{FromDate}/{ToDate}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /mytimecard/range/{FromDate}/{ToDate}/{userid}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /mytimecard/modifiedsince/{FromDate} /mytimecard/modifiedsince/{FromDate}/page/{pageindex}/{pagesize}/{orderby}/{ascending} </pre>
Verb	GET
Access	All
Description	<p>Get a list of timecards associated with this logged on account.</p> <p>The modifiedsince endpoint returns all the timecards that have been modified since the FromDate.</p>
Schema	https://api.springahead.com/v1/mytimecard/schema

<p>Example Response</p>	<pre> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecordCount>10</RecordCount> <PageIndex>1</PageIndex> <PageSize>1</PageSize> <PageCount>10</PageCount> <Timecards> <Timecard> <Id>2</Id> <CreateDate>2011-04-08T10:22:00</CreateDate> <CreatedUserId>0</CreatedUserId> <ModifiedDate>2011-04-08T10:24:00</ModifiedDate> <ModifiedUserId>0</ModifiedUserId> <Submit>>false</Submit> <SubmitDate>2011-04-08T10:22:00</SubmitDate> <SubmitUserId>0</SubmitUserId> <Row>0</Row> <UserId>0</UserId> <TimecardDate>2011-04-04T00:00:00</TimecardDate> <Projectid>0</Projectid> <Taskid>-1</Taskid> <TimeType>0</TimeType> <HoursAdjust>0</HoursAdjust> <HoursBreak>0</HoursBreak> <HoursDay>4</HoursDay> <ApprovedStatus>0</ApprovedStatus> <ApprovedUserId>0</ApprovedUserId> <DateApproved xsi:nil="true"/> <ProcessId>-1</ProcessId> <ProcessUserId>0</ProcessUserId> <ProcessDate xsi:nil="true"/> <BillableId>-1</BillableId> <BillableUserId>0</BillableUserId> <BillableDate xsi:nil="true"/> <PayableId>-1</PayableId> <PayableUserId>0</PayableUserId> <PayableDate xsi:nil="true"/> <InvoicedDate xsi:nil="true"/> <BillRateBase>0</BillRateBase> <BillRateMult>0</BillRateMult> <PayRateBase>0</PayRateBase> <PayRateMult>0</PayRateMult> <ItemId>0</ItemId> </Timecard> </Timecards> </SpringAhead> </pre>
<p>Known issues</p>	<p>Currently this is Read only.</p>

Expenses

URI Format	/expense, /expenses
Verb	GET
Description	Return all the expenses in the system.
Example Response	<pre><SpringAhead> </SpringAhead></pre>
Known issues	Not Implemented

InvoiceAccount

List

URI Format	<pre> /invoiceaccount /invoiceaccount/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoiceaccounts /invoiceaccounts/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoiceaccount/modifiedsince/{from} /invoiceaccount/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoiceaccounts/modifiedsince/{from} /invoiceaccounts/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoiceaccount/filter/{filterby}/{filtervalue} /invoiceaccount/filter/{filterby}/{filtervalue}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoiceaccounts/filter/{filterby}/{filtervalue} /invoiceaccounts/filter/{filterby}/{filtervalue}/page/{pageindex}/{pagesize}/{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. from - the Date to list from. Use Sql time convention to avoid localization issues. eg 2011-1-12T12:00:00 filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string. </pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoiceaccount/schema
Description	Return a list of all the Accounts. This is using the accounting definition of Account, so its not the "Accounts" of the logon users. That will be the <i>Users</i> endpoint.

<p>Example Response</p>	<pre> <SpringAhead xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <RecordCount>6</RecordCount> <PageIndex>0</PageIndex> <PageSize>100</PageSize> <PageCount>1</PageCount> <InvoiceAccounts> <InvoiceAccount> <Id>4</Id> <Active>true</Active> <CreatedDate>2011-07-26T18:31:34</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-07-26T18:31:34</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>123</Name> <Number>23</Number> <ErpAccountType>Accounts_Payable</ErpAccountType> </InvoiceAccount> <InvoiceAccount> <Id>5</Id> <Active>true</Active> <CreatedDate>2011-07-26T18:31:40</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-07-26T18:31:40</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>345</Name> <Number>345</Number> <ErpAccountType>Accounts_Receivable</ErpAccountType> </InvoiceAccount> <InvoiceAccount> <Id>2</Id> <Active>true</Active> <CreatedDate>2011-03-25T15:04:16</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-25T15:04:16</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>dsgsd</Name> <Number>43534</Number> <ErpAccountType>Cost_Of_Goods_Sold</ErpAccountType> </InvoiceAccount> <InvoiceAccount> <Id>1</Id> <Active>true</Active> <CreatedDate>2011-03-25T15:04:00</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-25T15:04:00</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Joe Blow</Name> <Number>233234</Number> <ErpAccountType>Income</ErpAccountType> </InvoiceAccount> <InvoiceAccount> <Id>6</Id> <Active>true</Active> <CreatedDate>2011-10-10T15:30:00</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-10-10T15:30:00</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>sdfsdas</Name> <Number>sdfsf</Number> <ErpAccountType>Income</ErpAccountType> <LastInvoiceNumber>dfg</LastInvoiceNumber> </InvoiceAccount> <InvoiceAccount> <Id>3</Id> <Active>true</Active> <CreatedDate>2011-03-29T17:16:59</CreatedDate> </pre>
-------------------------	---

```

    <CreatedUserId>1</CreatedUserId>
    <ModifiedDate>2011-03-29T17:16:59</ModifiedDate>
    <ModifiedUserId>1</ModifiedUserId>
    <Name>The Company Credit Card</Name>
    <Number>2342342354234</Number>
    <ErpAccountType>Credit_Card</ErpAccountType>
  </InvoiceAccount>
</InvoiceAccounts>
</SpringAhead>

```

Example using the filter;

<http://api.springahead.com/v1/InvoiceAccount/filter/Number/4>

```

<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <RecordCount>4</RecordCount>
  <PageIndex>0</PageIndex>
  <PageSize>100</PageSize>
  <PageCount>1</PageCount>
  <InvoiceAccounts>
    <InvoiceAccount>
      <Id>5</Id>
      <Active>true</Active>
      <CreateDate>2011-07-26T18:31:40</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-07-26T18:31:40</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <Name>345</Name>
      <Number>345</Number>
      <Type>Accounts_Receivable</Type>
    </InvoiceAccount>
    <InvoiceAccount>
      <Id>2</Id>
      <Active>true</Active>
      <CreateDate>2011-03-25T15:04:16</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-03-25T15:04:16</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <Name>d sdfgsd</Name>
      <Number>43534</Number>
      <Type>Cost_Of_Goods_Sold</Type>
    </InvoiceAccount>
    <InvoiceAccount>
      <Id>1</Id>
      <Active>true</Active>
      <CreateDate>2011-03-25T15:04:00</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-03-25T15:04:00</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <Name>Joe Blow</Name>
      <Number>233234</Number>
      <Type>Income</Type>
    </InvoiceAccount>
    <InvoiceAccount>
      <Id>3</Id>
      <Active>true</Active>
      <CreateDate>2011-03-29T17:16:59</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-03-29T17:16:59</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <Name>The Company Credit Card</Name>
      <Number>2342342354234</Number>
      <Type>Credit_Card</Type>
    </InvoiceAccount>
  </InvoiceAccounts>
</SpringAhead>

```

Known issues	

Details

URI Format	/invoiceaccount/{id} id - int
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoiceaccount/schema
Description	Return details about an Account.
Example Response	http://api.springahead.com/v1/invoiceaccount/5 <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceAccount> <Id>5</Id> <Active>true</Active> <CreateDate>2011-07-26T18:31:40</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-07-26T18:31:40</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>345</Name> <Number>345</Number> <Type>Accounts_Receivable</Type> </InvoiceAccount> </SpringAhead></pre>
Known issues	

Create

URI Format	/invoiceaccount
Verb	PUT
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoiceaccount/schema
Description	Create an Account, or multiple Accounts. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.

Example Request	http://api.springahead.com/v1/invoiceaccount <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceAccount> <Active>true</Active> <Name>345</Name> <Number>345</Number> <Type>Accounts_Receivable</Type> </InvoiceAccount> <InvoiceAccount> <Active>true</Active> <Name>456</Name> <Number>678</Number> <Type>Accounts_Receivable</Type> </InvoiceAccount> </SpringAhead></pre>
Known issues	

Edit

URI Format	/invoiceaccount /invoiceaccount/{id} id - int
Verb	POST
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoiceaccount/schema
Description	Edit an Account, or edit multiple Accounts. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.

<p>Example Request</p>	<p>http://api.springahead.com/v1/invoiceaccount/5</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceAccount> <Id>5</Id> <Active>>true</Active> <Name>345</Name> <Number>345</Number> <Type>Accounts_Receivable</Type> </InvoiceAccount> </SpringAhead></pre> <p>Multi edit - Edit both 5 and 7</p> <p>http://api.springahead.com/v1/invoiceaccount</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceAccount> <Id>5</Id> <Active>>true</Active> <Type>Accounts_Receivable</Type> </InvoiceAccount> <InvoiceAccount> <Id>7</Id> <Active>>true</Active> <Type>Accounts_Receivable</Type> </InvoiceAccount> </SpringAhead></pre>
<p>Known issues</p>	

Classification

List

URI Format	<pre> /classification /classifications /classification/page/{pageindex}/{pagesize}/{orderby}/{ascending} /classifications/page/{pageindex}/{pagesize}/{orderby}/{ascending} /classification/modifiedsince/{from} /classifications/modifiedsince/{from} /classification/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /classifications/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /classification/filter/{filterby}/{filtervalue} /classes/filter/{filterby}/{filtervalue} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. from - the Date to list from. Use Sql time convention to avoid localization issues. eg 2011-1-12T12:00:00 filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string. </pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/classification/schema
Description	Return a list of all the Class's.

<p>Example Response</p>	<pre> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecordCount>3</RecordCount> <PageIndex>0</PageIndex> <PageSize>100</PageSize> <PageCount>1</PageCount> <Classifications> <Classification> <Id>1</Id> <Active>true</Active> <CreateDate>2011-03-28T13:48:02</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-28T13:48:02</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Class Something</Name> </Classification> <Classification> <Id>3</Id> <Active>true</Active> <CreateDate>2011-03-28T13:48:27</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-28T13:48:27</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Fred Class</Name> </Classification> <Classificaiton> <Id>2</Id> <Active>true</Active> <CreateDate>2011-03-28T13:48:15</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-28T13:48:15</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Yo Class</Name> </Classification> </Classifications> </SpringAhead> Example using the filter; http://api.springahead.com/v1/class/filter/Name/some <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecordCount>1</RecordCount> <PageIndex>0</PageIndex> <PageSize>100</PageSize> <PageCount>1</PageCount> <Classification> <Classification> <Id>1</Id> <Active>true</Active> <CreateDate>2011-03-28T13:48:02</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-28T13:48:02</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Class Something</Name> </Classification> </Classifications> </SpringAhead> </pre>
<p>Known issues</p>	<p>Filter names are case sensitive.</p>

Details

URI Format	/classification/{id} /classifications/{id} id - int
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/classification/schema
Description	Return details about an Account.
Example Response	http://api.springahead.com/v1/classification/1 <pre> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Classification> <Id>1</Id> <Active>true</Active> <CreatedDate>2011-03-28T13:48:02</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-03-28T13:48:02</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Class Something</Name> </Classification> </SpringAhead> </pre>
Known issues	

Create

URI Format	/classification /classifications
Verb	PUT
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/classification/schema
Description	Create a Class, or multiple Classes. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.

Example Request	http://api.springahead.com/v1/classification <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Classification> <Active>true</Active> <Name>345</Name> </Classification> <Classification> <Active>true</Active> <Name>456</Name> </Classification> </SpringAhead></pre>
Known issues	

Edit

URI Format	<pre>/classification /classification/{id} /classifications /classifications/{id}</pre> <p>id - int</p>
Verb	POST
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/classification/schema
Description	Edit a Class, or edit multiple Classes. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.

<p>Example Request</p>	<p>http://api.springahead.com/v1/classification/5</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Classification> <Id>5</Id> <Active>true</Active> <Name>345</Name> </Classification> </SpringAhead></pre> <p>Multi edit - Edit both 5 and 7</p> <p>http://api.springahead.com/v1/classification</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Classification> <Id>5</Id> <Active>false</Active> </Classification> <Classification> <Id>7</Id> <Active>false</Active> </Classification> </SpringAhead></pre>
<p>Known issues</p>	

Company

List

URI Format	<pre> /company /companies /company/page/{pageindex}/{pagesize}/{orderby}/{ascending} /companies/page/{pageindex}/{pagesize}/{orderby}/{ascending} /company/modifiedsince/{from} /companies/modifiedsince/{from} /company/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /companies/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /company/filter/{filterby}/{filtervalue} /companys/filter/{filterby}/{filtervalue} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. from - the Date to list from. Use Sql time convention to avoid localization issues. eg 2011-1-12T12:00:00 filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string.</pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/company/schema
Description	Return a list of all the Companies.

<p>Example Response</p>	<p>http://api.springahead.com/v1/Company/page/0/2</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecordCount>18</RecordCount> <PageIndex>0</PageIndex> <PageSize>2</PageSize> <PageCount>9</PageCount> <Companies> <Company> <Id>17</Id> <Active>true</Active> <CreateDate>2011-07-07T11:21:59</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-07-07T11:21:59</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>aaa</Name> <RoleCustomer>false</RoleCustomer> <RoleVendor>false</RoleVendor> <BillCycle>All</BillCycle> <InvoiceMessage xsi:nil="true"/> <AccountsPayableContact xsi:nil="true"/> <InvoiceTerm xsi:nil="true"/> <FirstDayOfWeek>0</FirstDayOfWeek> <FirstDayOfMonth>0</FirstDayOfMonth> </Company> <Company> <Id>7</Id> <Active>true</Active> <CreateDate>2011-05-19T18:13:01</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-05-19T18:13:01</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>asd</Name> <RoleCustomer>false</RoleCustomer> <RoleVendor>false</RoleVendor> <BillCycle>All</BillCycle> <InvoiceMessage xsi:nil="true"/> <AccountsPayableContact xsi:nil="true"/> <InvoiceTerm xsi:nil="true"/> <FirstDayOfWeek>0</FirstDayOfWeek> <FirstDayOfMonth>0</FirstDayOfMonth> </Company> </Companies> </SpringAhead></pre>
<p>Known issues</p>	<p>Filter names are case sensitive.</p>

Details

<p>URI Format</p>	<pre>/company/{id} /companies/{id} id - int</pre>
<p>Verb</p>	<p>GET</p>
<p>Access</p>	<p>Admin, ApiAdmin</p>
<p>Schema</p>	<p>http://api.springahead.com/v1/company/schema</p>

Description	Return details about an Company.
Example Response	<pre> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Company> <Id>1</Id> <Active>true</Active> <CreatedDate>2011-03-23T23:00:25</CreatedDate> <CreatedUserId>2</CreatedUserId> <ModifiedDate>2011-04-21T18:07:14</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>Mindjet dfkkj</Name> <BillAddress> <Name>fd</Name> </BillAddress> <Phone1>+1 (03) 97266657</Phone1> <RoleCustomer>true</RoleCustomer> <RoleVendor>true</RoleVendor> <BillCycle>All</BillCycle> <InvoiceMessage xsi:nil="true"/> <AccountsPayableContact xsi:nil="true"/> <InvoiceTerm xsi:nil="true"/> <FirstDayOfWeek>0</FirstDayOfWeek> <FirstDayOfMonth>0</FirstDayOfMonth> </Company> </SpringAhead> </pre>
Known issues	

Create

URI Format	/company /companies
Verb	PUT
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/account/schema
Description	Create a Company, or multiple Companies. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.

<p>Example Request</p>	<p>http://api.springahead.com/v1/class</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Company> <Active>true</Active> <Name>Some company name</Name> <BillAddress> <Name>Some company address</Name> </BillAddress> <Phone1>+1 (03) 97266657</Phone1> <RoleCustomer>true</RoleCustomer> <RoleVendor>true</RoleVendor> <BillCycle>All</BillCycle> <FirstDayOfWeek>0</FirstDayOfWeek> <FirstDayOfMonth>0</FirstDayOfMonth> </Company> <Company> <Active>true</Active> <Name>Some other company name</Name> <BillAddress> <Name>Some company address</Name> </BillAddress> <Phone1>+1 (03) 97266657</Phone1> <RoleCustomer>true</RoleCustomer> <RoleVendor>true</RoleVendor> <BillCycle>All</BillCycle> <FirstDayOfWeek>0</FirstDayOfWeek> <FirstDayOfMonth>0</FirstDayOfMonth> </Company> </SpringAhead></pre>
<p>Known issues</p>	

Edit

<p>URI Format</p>	<p>/company /companies /company/{id} /companies/{id}</p> <p>id - int</p>
<p>Verb</p>	<p>POST</p>
<p>Access</p>	<p>Admin, ApiAdmin</p>
<p>Schema</p>	<p>http://api.springahead.com/v1/company/schema</p>
<p>Description</p>	<p>Edit a Company, or edit multiple Companies. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.</p>

Example Request	<p>http://api.springahead.com/v1/company/5</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Company> <Phone1>+1 (03) 97266657</Phone1> </Company> </SpringAhead></pre> <p>Multi edit - Edit both 5 and 7</p> <p>http://api.springahead.com/v1/company</p> <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Company> <Id>5</Id> <Active>>false</Active> </Company> <Company> <Id>7</Id> <Active>>false</Active> </Company> </SpringAhead></pre>
Known issues	

MyCompany / Enterprise

List

URI Format	/mycompany /enterprise
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/mycompany/schema
Description	Return the MyCompany information.

Example Response

```
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Enterprise>
    <Id>1</Id>
    <Active>true</Active>
    <CreateDate>2011-03-10T10:53:25</CreateDate>
    <CreatedUserId>1</CreatedUserId>
    <ModifiedDate>2011-08-10T14:07:13</ModifiedDate>
    <ModifiedUserId>1</ModifiedUserId>
    <Name>Demo Admin</Name>
    <LoginName>Demo Admin</LoginName>
    <TimeZone>PST</TimeZone>
    <Currency>USD_US_dollar</Currency>
    <CurrencySymbol>$</CurrencySymbol>
    <FirstDayOfWeek>0</FirstDayOfWeek>
    <FirstDayOfMonth xsi:nil="true"/>
    <EmailNotificationsEnabled>true</EmailNotificationsEnabled>
    <RetainHistoryTime xsi:nil="true"/>
    <EnableBrand xsi:nil="true"/>
    <BillingHidden xsi:nil="true"/>
    <NameHidden xsi:nil="true"/>
    <AccountType>Free_Trial</AccountType>
    <RolePartner xsi:nil="true"/>
    <Redirect xsi:nil="true"/>
    <UseTasks>true</UseTasks>
    <UseUserRates>true</UseUserRates>
    <UseUserReps>true</UseUserReps>
    <ClassMandatory>false</ClassMandatory>
    <AccountsReceivableAccountMandatory>false</AccountsReceivableAccountMandatory>
    <SyncExpenseAccounts xsi:nil="true"/>
    <SyncInventoryItems xsi:nil="true"/>
    <DateTimeAllowed xsi:nil="true"/>
    <IdProcess>0</IdProcess>
    <IdBillable>0</IdBillable>
    <IdPayable>0</IdPayable>
    <AttachmentsEnabled xsi:nil="true"/>
    <SetupWizardStatus>true</SetupWizardStatus>
    <LogoutTimeout xsi:nil="true"/>
    <TrialProps>
      <Id>2</Id>
      <Active>true</Active>
      <CreateDate>2011-07-18T17:09:16</CreateDate>
      <CreatedUserId>0</CreatedUserId>
      <ModifiedDate>2011-07-18T17:09:16</ModifiedDate>
      <ModifiedUserId>0</ModifiedUserId>
      <ExpirationDate>2012-08-01T17:09:16</ExpirationDate>
      <DemoDataUsed>false</DemoDataUsed>
      <NumSyncs>0</NumSyncs>
      <NumTrialResets>0</NumTrialResets>
      <DunningState>0</DunningState>
      <NumLogins>0</NumLogins>
    </TrialProps>
    <Expense>
      <Enabled>true</Enabled>
      <Id>1</Id>
      <UseForeignExchange xsi:nil="true"/>
      <UseCreditCardDownload xsi:nil="true"/>
      <OffsetAccount>
        <Id>1</Id>
        <Active>true</Active>
        <CreateDate>2011-03-25T15:04:00</CreateDate>
        <CreatedUserId>1</CreatedUserId>
        <ModifiedDate>2011-03-25T15:04:00</ModifiedDate>
        <ModifiedUserId>1</ModifiedUserId>
        <Name>Joe Blow</Name>
        <Number>233234</Number>
        <Type>Income</Type>
      </OffsetAccount>
    </Expense>
  </Enterprise>
</SpringAhead>
```

```

</OffsetAccount>
<ExpenseRollup xsi:nil="true"/>
<ExpenseDetailRollup xsi:nil="true"/>
<ExpenseToPayroll xsi:nil="true"/>
</Expense>
<AccountingExport>
  <Software>NotSet</Software>
  <BillByItem>>false</BillByItem>
  <Pay1099Check xsi:nil="true"/>
  <TimeDateType>1</TimeDateType>
  <ExpenseDateType>0</ExpenseDateType>
  <CreditChargeDateType>50</CreditChargeDateType>
</AccountingExport>
<TimecardInfo>
  <Enabled>>true</Enabled>
  <DefaultNewAsSubmitted>>false</DefaultNewAsSubmitted>
  <InitialRows>0</InitialRows>
  <NewRowsAfterSaving>0</NewRowsAfterSaving>
  <AllowNewTimeAfterApproval>>true</AllowNewTimeAfterApproval>
  <UseTimeInTimeOut>>true</UseTimeInTimeOut>
  <TimeIncrementMinutes xsi:nil="true"/>
  <UseClockInClockOut xsi:nil="true"/>
  <ClockMinutesRound xsi:nil="true"/>
  <ClockMinutesLate xsi:nil="true"/>
  <ChangesRequireReason xsi:nil="true"/>
  <Cycle xsi:nil="true"/>
  <UseAccrual>>true</UseAccrual>
  <AccrualDate>2011-02-01T00:00:00</AccrualDate>
  <AccrualInDays>>false</AccrualInDays>
  <Overtime>
    <OvertimeDayHours xsi:nil="true"/>
    <OvertimeWeekHours>40</OvertimeWeekHours>
    <DoubleTimeDayHours xsi:nil="true"/>
    <Enforce xsi:nil="true"/>
  </Overtime>
</TimecardInfo>
<PayrollItem>
  <Regular>2</Regular>
  <Overtime xsi:nil="true"/>
  <Doubletime xsi:nil="true"/>
  <Sick xsi:nil="true"/>
  <Vacation xsi:nil="true"/>
  <Holiday xsi:nil="true"/>
  <NonBill xsi:nil="true"/>
  <SalaryRegular>1</SalaryRegular>
  <SalaryOvertime xsi:nil="true"/>
  <SalaryDoubletime xsi:nil="true"/>
  <SalarySick xsi:nil="true"/>
  <SalaryVacation xsi:nil="true"/>
  <SalaryHoliday xsi:nil="true"/>
  <SalaryNonBill xsi:nil="true"/>
  <Enabled xsi:nil="true"/>
</PayrollItem>
<PayrollExport>
  <Software>NotSet</Software>
  <UseDepartmentCodes xsi:nil="true"/>
</PayrollExport>
<Payables>
  <Software>NotSet</Software>
  <Pay1099W2 xsi:nil="true"/>
  <PayExpensePayroll xsi:nil="true"/>
  <Enabled>>true</Enabled>
  <PayrollRollup>0</PayrollRollup>
  <Burden xsi:nil="true"/>
</Payables>
<Invoicing>
  <Enabled>>true</Enabled>

```

	<pre> <ExportTimecards>false</ExportTimecards> <ExportInvoices>true</ExportInvoices> <LastNameThenFirst>true</LastNameThenFirst> <RepDisabled>true</RepDisabled> <RepUseInitials>true</RepUseInitials> <ExportTimeWithAccounting xsi:nil="true"/> <ExportTimeWithPayroll xsi:nil="true"/> <ExportPayroll>false</ExportPayroll> <ExportPayables>true</ExportPayables> <Cycle>0</Cycle> <IncludeTimeDescriptions>false</IncludeTimeDescriptions> <InvoiceDate>0</InvoiceDate> <Software xsi:nil="true"/> <SalesTax>true</SalesTax> <UseItemDescriptionForBilling>false</UseItemDescriptionForBilling> </Invoicing> <Approval> <UseTimeApprovers>true</UseTimeApprovers> <UseExpenseApprovers>true</UseExpenseApprovers> <ApprovalAnyTime>true</ApprovalAnyTime> <ApprovalAnyExpense>true</ApprovalAnyExpense> <UseApprovalLevels>true</UseApprovalLevels> <ExpenseApprover> <Id>4</Id> <Active>true</Active> <CreateDate>2011-04-01T12:06:03</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-08-11T16:38:54</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <LoginName>cox</LoginName> <FirstName>Mike</FirstName> <MiddleInitial>F</MiddleInitial> <LastName>Cox</LastName> <Initials>MC</Initials> </ExpenseApprover> </Approval> </Enterprise> </SpringAhead> </pre>
Known issues	This is read only at the moment.

ExpenseType

Item

InvoiceMessage

List

URI Format	<pre> /invoicemessage /invoicemessage/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoicemessages /invoicemessages/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoicemessage/modifiedsince/{from} /invoicemessage/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoicemessages/modifiedsince/{from} /invoicemessages/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoicemessage/filter/{filterby}/{filtervalue} /invoicemessage/filter/{filterby}/{filtervalue}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /invoicemessages/filter/{filterby}/{filtervalue} /invoicemessages/filter/{filterby}/{filtervalue}/page/{pageindex}/{pagesize}/{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. from - the Date to list from. Use Sql time convention to avoid localization issues. eg 2011-1-12T12:00:00 filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string. </pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoicemessage/schema
Description	Return a list of all the InvoiceMessages.

Example Response	<pre> <?xml version="1.0" encoding="utf-8"?> <SpringAhead xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <RecordCount>8</RecordCount> <PageIndex>0</PageIndex> <PageSize>3</PageSize> <PageCount>3</PageCount> <InvoiceMessages> <InvoiceMessage> <Id>37</Id> <Active>true</Active> <CreatedDate>2011-05-11T13:26:38</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-05-11T16:06:49</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>123ddsdfdfsv</Name> </InvoiceMessage> <InvoiceMessage> <Id>38</Id> <Active>true</Active> <CreatedDate>2011-05-11T13:27:48</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-05-11T13:27:48</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>123df</Name> </InvoiceMessage> <InvoiceMessage> <Id>35</Id> <Active>true</Active> <CreatedDate>2011-04-20T14:16:36</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-04-20T14:16:55</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>1d</Name> </InvoiceMessage> </InvoiceMessages> </SpringAhead> </pre>
Known issues	

Details

URI Format	/invoicemessage/{id} id - int
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoicemessage/schema
Description	Return details about an InvoiceMessage.

Example Response	http://api.springahead.com/v1/invoicemessage/37 <pre><?xml version="1.0" encoding="utf-8"?> <SpringAhead xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <Message> <Id>37</Id> <Active>true</Active> <CreatedDate>2011-05-11T13:26:38</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-05-11T16:06:49</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>123ddsdfdfsv</Name> </Message> </SpringAhead></pre>
Known issues	

Create

URI Format	/invoicemessage
Verb	PUT
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/invoicemessage/schema
Description	Create an InvoiceMessage, or multiple InvoiceMessages. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.
Example Request	http://api.springahead.com/v1/invoicemessage <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceMessage> <Active>true</Active> <Name>The API service gives you access.</Name> </InvoiceMessage> <InvoiceMessage> <Active>true</Active> <Name>To the InvoiceMessages.</Name> </InvoiceMessage> </SpringAhead></pre>
Known issues	

Edit

URI Format	/invoicemessage /invoicemessage/{id} id - int
Verb	POST
Access	Admin, ApiAdmin

Schema	http://api.springahead.com/v1/invoicemessage/schema
Description	Edit an InvoiceMessage, or edit multiple InvoiceMessages. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.
Example Request	http://api.springahead.com/v1/invoicemessage/5 <pre><SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <InvoiceMessage> <Id>5</Id> <Active>true</Active> <Name>Some other name</Name> </InvoiceMessage> </SpringAhead></pre>
Known issues	

PaymentMethod

PayrollItem

Project

List

URI Format	<pre> /project /projects /project/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/page/{pageindex}/{pagesize}/{orderby}/{ascending} /project/modifiedsince/{from} /projects/modifiedsince/{from} /project/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/modifiedsince/{from}/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /project/filter/{filterby}/{filtervalue} /projects/filter/{filterby}/{filtervalue} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. from - the Date to list from. Use Sql time convention to avoid localization issues. eg 2011-1-12T12:00:00 filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string.</pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/project/schema
Description	Return a list of all the projects..
Example Response	http://api.springahead.com/v1/project/page/0/2
Known issues	Filter names are case sensitive.

Details

URI Format	<pre> /project/{id} /projects/{id} id - int</pre>
Verb	GET

Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/project/schema
Description	Return details about an Project.
Example Response	
Known issues	

Create

URI Format	/project /projects
Verb	PUT
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/project/schema
Description	Create a Project, or multiple Projects. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.
Example Request	http://api.springahead.com/v1/project
Known issues	Project names must conform to the following naming convention. <company name>:<project name>

Edit

URI Format	/project /projects /project/{id} /projects/{id} id - int
Verb	POST
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/project/schema
Description	Edit a Project, or edit multiple Projects. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.
Example Request	http://api.springahead.com/v1/project/5 Multi edit - Edit both 5 and 7

Known issues	Project names must conform to the following naming convention. <company name>:<project name>
--------------	---

Task

TODO: Similar to others

Term

TODO: Similar to others

TimeType

TODO: Similar to others

User

List

URI Format	<pre> /user /users /user/page/{pageindex}/{pagesize}/{orderby}/{ascending} /users/page/{pageindex}/{pagesize}/{orderby}/{ascending} /user/filter/{filterby}/{filtervalue} /users/filter/{filterby}/{filtervalue} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool filterby - the XML name to filter the results on. filtervalue - the search string to use. This will match anywhere in the string.</pre>
Verb	GET
Access	Admin, ApiAdmin
Schema	http://api.springahead.com/v1/user/schema
Description	Return a list of all the Users.
Example	http://api.springahead.com/v1/user/page/0/2
Known issues	Filter names are case sensitive.

Details

URI Format	<pre> /user/{id} /users/{id} id - int</pre>
Verb	GET
Access	Admin, ApiAdmin
Description	Return details about an User.
Example Response	
Known issues	

CurrentUser

URI Format	/currentuser/{id} /myuser/{id} id - int
Verb	GET
Access	All
Description	Return details about the current User.
Example Response	
Known issues	

Create

URI Format	/user /users
Verb	PUT
Access	Admin, ApiAdmin
Description	Create a User, or multiple Users. You don't need to specify the Id, it'll be ignored. The created Id's will be returned in the response.
Example Request	http://api.springahead.com/v1/user
Known issues	

Edit

URI Format	/user /users /user/{id} /users/{id} id - int
Verb	POST
Access	Admin, ApiAdmin
Description	Edit a User, or edit multiple Users. You don't need to specify all the XML elements, just the ones you want to change will be sufficient. Others will not be changed.
Example Request	http://api.springahead.com/v1/user/5 Multi edit - Edit both 5 and 7
Known issues	

Xid

The Xid tables are used to map unique identifiers in the SpringAhead system to unique identifiers in other systems. Maybe yours?

The Xid table specifies a SystemGuid. This is a Guid that represents the system that this Xid is mapping to. We have some already predefined, but it will accept new ones. Just make sure your don't change it.

List

URI Format	<pre> /xid /xid/page/{pageindex}/{pagesize}/{orderby}/{ascending} /xid/name/{search} /xid/name/{search}/page/{pageindex}/{pagesize}{orderby}/{ascending} /xid/xid/{search} /xid/xid/{search}/page/{pageindex}/{pagesize}{orderby}/{ascending} /xid/table/{search} /xid/table/{search}/page/{pageindex}/{pagesize}{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </pre>
Verb	GET
Access	Admin, ApiAdmin
Description	<p>Return a list of all the XID's associated with this enterprise.</p> <p>The <code>/xid/name/{search}</code> api searches the xids and returns ones that match the <i>name</i> field.</p> <p>The <code>/xid/xid/{search}</code> api searches the xids and returns ones that match the <i>Xid</i> field.</p> <p>The <code>/xid/table/{search}</code> api searches the xids and returns ones that match the <i>table</i> field.</p>

<p>Example Response</p>	<pre><Xids xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecordCount>2</RecordCount> <PageIndex>1</PageIndex> <PageSize>100</PageSize> <PageCount>1</PageCount> <List> <Xid> <Id>38</Id> <Active>true</Active> <XId>1</XId> <CreateDate>2011-04-27T17:58:00</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-04-27T17:58:00</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <TableName>Expense</TableName> <ObjectId>1</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>123456789</Name> </Xid> <Xid> <Id>31</Id> <Active>true</Active> <XId>1</XId> <CreateDate>2011-04-27T15:48:00</CreateDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-04-27T15:48:00</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <TableName>Expense</TableName> <ObjectId>1</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>12345678901234567890</Name> </Xid> </List> </Xids></pre>
<p>Known issues</p>	<p>This can be like drinking from a fire hose.</p>

Details

URI Format	/xid/{id}
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific XID associated with this enterprise.

Example Response	<pre> Example Xml for /Xid/31 <SpringAhead> <Xid> <Id>31</Id> <Active>>false</Active> <Xid>123</Xid> <CreatedDate>2011-04-27T15:48:00</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-05-05T23:03:00</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <TableName>User</TableName> <ObjectId>2</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>1234dfs5678901234567890</Name> </Xid> </SpringAhead> </pre>
Known issues	

Edit

URI Format	/xid/{id}
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an XId.
Example Request	<pre> Example Xml for /Xid/31, set active to true <SpringAhead> <Xid> <Active>>true</Active> </Xid> </SpringAhead> </pre>
Known issues	

MultiEdit

URI Format	/xid
Verb	POST
Description	Update specific settings on an XId.

Example Request	<p>Example Xml for 31 and 32, set active to true on both</p> <pre><SpringAhead> <Xid> <Id>31</Id> <Active>true</Active> </Xid> <Xid> <Id>32</Id> <Active>true</Active> </Xid> </SpringAhead></pre>
Known issues	

Create

URI Format	/xid
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Xid entries. This API supports multiple Xid's in the Request. So you don't have to call over and over again to create new ones.
Example Request	<pre><?xml version="1.0" encoding="utf-8" ?> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Xid> <Active>true</Active> <XId>1</XId> <TableName>Expense</TableName> <ObjectId>1</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>Test1</Name> </Xid> <Xid> <Active>true</Active> <XId>2</XId> <TableName>Expense</TableName> <ObjectId>1</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>Test2</Name> </Xid> <Xid> <Active>true</Active> <XId>3</XId> <TableName>Expense</TableName> <ObjectId>1</ObjectId> <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid> <Name>Test3</Name> </Xid> </SpringAhead></pre> <p>Sending this XML will cause 3 new XID's to be created.</p>
Known issues	

Delete

URI Format	/xid /xid/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	<p>Delete one Xid or multiple Xid entries.</p> <p>This API supports multiple Xid's in the Request. So you don't have to call over and over again to delete entities.</p> <p>Technically with REST we're not supposed to send data on DELETE verbs, but its useful in this case.</p>
Example Request	<pre><?xml version="1.0" encoding="utf-8" ?> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <Xid> <XId>1</XId> </Xid> <Xid> <XId>2</XId> </Xid> <Xid> <XId>3</XId> </Xid> </SpringAhead></pre> <p>Sending this XML will cause 3 XID's to be deleted.</p> <p>Sending a DELETE on /Xid/23 will delete entry with id 23.</p>
Known issues	

Search

	<pre>/Xid/Name/{search} /Xid/Name/{search}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /Xid/Xid/{search} /Xid/Xid/{search}/page/{pageindex}/{pagesize}/{orderby}/{ascending} /Xid/table/{tablename}/{objectid} /Xid/table/{tablename}/{objectid}/page/{pageindex}/{pagesize}/{orderby}/{ascending}</pre>
VERB	GET
Access	Admin, ApiAdmin
Description	Sometimes you need to find a particular name or Xid. These endpoints allow you to get a list of all Names that match <i>search</i> . Or all Xids that match <i>search</i> .

Example Response

<http://api.springahead.com/v1/Xid/Table/User/1>

```
<SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <RecordCount>4</RecordCount>
  <PageIndex>0</PageIndex>
  <PageSize>100</PageSize>
  <PageCount>1</PageCount>
  <Xids>
    <Xid>
      <Id>46</Id>
      <Active>true</Active>
      <Xid>1ds</Xid>
      <CreateDate>2011-04-28T17:30:00</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-04-28T17:34:00</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <TableName>User</TableName>
      <ObjectId>1</ObjectId>
      <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid>
      <Name>sdfTestdfgdfghfghgdfdsfgdfgdfgdfgdfgdfg</Name>
    </Xid>
    <Xid>
      <Id>49</Id>
      <Active>true</Active>
      <Xid>1dsfgsdddffdsfdss</Xid>
      <CreateDate>2011-04-29T11:21:00</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-04-29T11:21:00</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <TableName>User</TableName>
      <ObjectId>1</ObjectId>
      <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid>
      <Name>sdfTestsfdfgdfgdfgdfgdfgdfgdfg</Name>
    </Xid>
    <Xid>
      <Id>47</Id>
      <Active>true</Active>
      <Xid>1dsfgsfdss</Xid>
      <CreateDate>2011-04-29T09:25:00</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-04-29T09:25:00</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <TableName>User</TableName>
      <ObjectId>1</ObjectId>
      <SystemGuid>807734da-2e70-e011-bb59-5c260a2cd941</SystemGuid>
      <Name>sdfTestsfdfgdfgdfgdfgdfgdfgdfg</Name>
    </Xid>
    <Xid>
      <Id>4</Id>
      <Active>true</Active>
      <Xid>{8C365670-A07D-4F74-B989-C02412CE0A62}</Xid>
      <CreateDate>2011-04-26T18:00:00</CreateDate>
      <CreatedUserId>1</CreatedUserId>
      <ModifiedDate>2011-04-26T18:00:00</ModifiedDate>
      <ModifiedUserId>1</ModifiedUserId>
      <TableName>User</TableName>
      <ObjectId>1</ObjectId>
      <SystemGuid>3b8ffcbb-7ac4-46d0-aa73-80a3f7b37e88</SystemGuid>
      <Name>CreateThisXid</Name>
    </Xid>
  </Xids>
</SpringAhead>
```


Known Issues	None
--------------	------

Secondaries

We have data associated with both Users and Projects that we refer to as Secondaries. These include Rates, Reps, POs, and Budgets on Projects, Rates and Reps on Users.

Secondaries for Projects

Rates

Rates can be found on the Project page in the Settings area. Here is an example of what some Rates look like on SpringAhead. Beth Berman has a bill rate of \$60 and a pay rate of \$35.

Settings / Projects / ABC Corp:Project A ▾ Active: Yes

Access
 Track time for ▾
 Track expenses for ▾

Project Payables
 This project is not payable
 This project is payable

Project Rates
 Use the billing rate for each person and task Bill everyone \$ per hour
 Use the pay rate for each person and task Pay everyone \$ per hour

People

Name	Active Dates	Bill Rate	Pay rate	Invoice Item
Berman, Beth	9/1/2011 - Current	\$60.00	\$35.00	Director
Adler, Alan	9/1/2011 - Current	\$70.00	\$50.00	IT - Network Configuration

List

URI Format	<pre> /project/{projectid}/rate /project/{projectid}/rate/page/{pageindex}/{pagesize}/{orderby}/{ascending} /project/{projectid}/rates /project/{projectid}/rates/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/rate /projects/{projectid}/rate/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/rates /projects/{projectid}/rates/page/{pageindex}/{pagesize}/{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool </pre>
------------	--

	search - the value being searched for.
Verb	GET
Access	Admin, ApiAdmin
Description	Return a list of all the Rates associated with the project specified.

<p>Example Response</p>	<pre> <?xml version="1.0" encoding="UTF-8"?> <SpringAhead xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance"> <RecordCount>3</RecordCount> <PageIndex>0</PageIndex> <PageSize>100</PageSize> <PageCount>1</PageCount> <ProjectRates> <ProjectRate> <Id>1502</Id> <DateStart>2011-12-06T00:00:00</DateStart> <DateStop xsi:nil="true" /> <User> <Id>5</Id> <Active>true</Active> <CreatedDate>2011-06-29T14:01:06</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-09-13T10:37:26</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <LastName>dsf</LastName> <EmployeeType>Salary</EmployeeType> </User> <BillRate>2</BillRate> <PayRate>2</PayRate> <Item> <Id>7</Id> <Active>true</Active> <CreatedDate>2011-12-06T18:54:55</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-12-06T18:54:55</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>3</Name> <UseDescriptionForBilling>false</UseDescriptionForBilling> </Item> </ProjectRate> <ProjectRate> <Id>1503</Id> <DateStart>2011-12-05T00:00:00</DateStart> <DateStop xsi:nil="true" /> <User> <Id>18</Id> <Active>true</Active> <CreatedDate>2011-08-08T17:28:48</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-08-08T17:28:55</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <LastName>dsf (Copy 10)</LastName> <EmployeeType>Salary</EmployeeType> </User> <BillRate>6</BillRate> <PayRate>5</PayRate> <Item> <Id>6</Id> <Active>true</Active> <CreatedDate>2011-12-06T18:54:45</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-12-06T19:09:16</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>2</Name> <UseDescriptionForBilling>false</UseDescriptionForBilling> </Item> </ProjectRate> </ProjectRates> </SpringAhead> </pre>
<p>Known issues</p>	

Details

URI Format	/project/{projectid}/rate/{id} /project/{projectid}/rates/{id} /projects/{projectid}/rate/{id} /projects/{projectid}/rates/{id}
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific Rate associated with the specified Project.
Example Response	<p>Example Xml for /v1/projects/61/rates/1502</p> <pre><SpringAhead xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <ProjectRate> <Id>1502</Id> <DateStart>2011-12-06T00:00:00</DateStart> <DateStop xsi:nil="true"/> <User> <Id>5</Id> <Active>true</Active> <CreatedDate>2011-06-29T14:01:06</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-09-13T10:37:26</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <LastName>dsf</LastName> <EmployeeType>Salary</EmployeeType> </User> <BillRate>2</BillRate> <PayRate>2</PayRate> <Item> <Id>7</Id> <Active>true</Active> <CreatedDate>2011-12-06T18:54:55</CreatedDate> <CreatedUserId>1</CreatedUserId> <ModifiedDate>2011-12-06T18:54:55</ModifiedDate> <ModifiedUserId>1</ModifiedUserId> <Name>3</Name> <UseDescriptionForBilling>false</UseDescriptionForBilling> </Item> </ProjectRate> </SpringAhead></pre>
Known issues	

Edit

URI Format	/project/{projectid}/rate/{id} /project/{projectid}/rates/{id} /projects/{projectid}/rate/{id} /projects/{projectid}/rates/{id}
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an Rate.

Example Request	<p>Example Xml for /project/1/rate/31, set active to true</p> <pre><SpringAhead> <ProjectRate> <Active>true</Active> <BillRate>120</BillRate> </ProjectRate> </SpringAhead></pre>
Known issues	

MultiEdit

URI Format	<pre>/project/{projectid}/rate /project/{projectid}/rates /projects/{projectid}/rate /projects/{projectid}/rates</pre>
Verb	POST
Description	Update specific settings on an XId.
Example Request	<p>Example Rate for 31 and 32, set active to true and BillRate on both</p> <pre><SpringAhead> <ProjectRate> <Id>31</Id> <Active>true</Active> <BillRate>110</BillRate> </ProjectRate> <ProjectRate> <Id>32</Id> <Active>true</Active> <BillRate>120</BillRate> </ProjectRate> </SpringAhead></pre>
Known issues	

Create

URI Format	<pre>/project/{projectid}/rate /project/{projectid}/rates /projects/{projectid}/rate /projects/{projectid}/rates</pre>
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Rate entries. This API supports multiple Rates in the Request. So you don't have to call over and over again to create new ones.

Example Request	<pre><?xml version="1.0" encoding="utf-8" ?> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <ProjectRate> <Active>true</Active> <DateStart>2011-1-1T00:00:00</DateStart> <DateStop>2011-12-31T00:00:00</DateStop> <User> <Id>1</Id> </User> <BillRate>120</BillRate> </ProjectRate> <ProjectRate> <Active>true</Active> <DateStart>2012-1-1T00:00:00</DateStart> <User> <Id>1</Id> </User> <BillRate>140</BillRate> </ProjectRate> </SpringAhead></pre> <p>Sending this XML will cause 2 new Rates to be created.</p>
Known issues	

Delete

URI Format	<pre>/project/{projectid}/rate/{id} /project/{projectid}/rates/{id} /projects/{projectid}/rate/{id} /projects/{projectid}/rates/{id}</pre>
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the rate specified by the id.
Known issues	

MultiDelete

URI Format	<pre>/project/{projectid}/rate /project/{projectid}/rates /projects/{projectid}/rate /projects/{projectid}/rates</pre>
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Rates specified by the list of ids in the request XML.

Example Request	<pre><SpringAhead> <ProjectRate> <Id>31</Id> </ProjectRate> <ProjectRate> <Id>32</Id> </ProjectRate> </SpringAhead></pre>
Known issues	

Reps

Reps can be found on the Project page in the Settings area, on the Approvers tab. Here is an example of what some Reps look like on SpringAhead.

Settings / Projects / ABC Corp:Project A Active: Yes

Overview Details Budget Invoicing People **Approvers** Tasks Time Settings


Manager Manager Manager approves time

Time Approvers Any approver can approve time All approvers must approve time

Only use time approvers on this page

Project Burden When calculating commissions, add a % burden to this project's pay rate

Reps

Name	Active Dates	Rep Type	Commission
Milbourne, Mark	3/5/2012 - Current	Account Manager	4000 

< People Cancel Save Save and Exit Tasks >

List

URI Format	<pre>/project/{projectid}/rep /project/{projectid}/rep/page/{pageindex}/{pagesize}/{orderby}/{ascending}</pre>
------------	--

	<pre> /project/{projectid}/reps /project/{projectid}/reps/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/rep /projects/{projectid}/rep/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/reps /projects/{projectid}/reps/page/{pageindex}/{pagesize}/{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </pre>
Verb	GET
Access	Admin, ApiAdmin
Description	Return a list of all the Reps associated with the project specified.
Example Response	
Known issues	

Details

URI Format	<pre> /project/{projectid}/rep/{id} /project/{projectid}/reps/{id} /projects/{projectid}/rep/{id} /projects/{projectid}/reps/{id} </pre>
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific Rep associated with the specified Project.
Example Response	
Known issues	

Edit

URI Format	<pre> /project/{projectid}/rep/{id} /project/{projectid}/reps/{id} /projects/{projectid}/rep/{id} /projects/{projectid}/reps/{id} </pre>
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an Rep.

Example Request	
Known issues	

MultiEdit

URI Format	/project/{projectid}/rep /project/{projectid}/reps /projects/{projectid}/rep /projects/{projectid}/reps
Verb	POST
Description	Update specific settings on a Rep.
Example Request	
Known issues	

Create

URI Format	/project/{projectid}/rep /project/{projectid}/reps /projects/{projectid}/rep /projects/{projectid}/reps
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Rep entries. This API supports multiple Reps in the Request. So you don't have to call over and over again to create new ones.
Example Request	
Known issues	

Delete

URI Format	/project/{projectid}/rep/{id} /project/{projectid}/reps/{id} /projects/{projectid}/rep/{id} /projects/{projectid}/reps/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the rep specified by the id.
Example Request	
Known issues	

MultiDelete

URI Format	/project/{projectid}/rep /project/{projectid}/reps /projects/{projectid}/rep /projects/{projectid}/reps
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Reps specified by the list of ids in the request XML.
Example Request	
Known issues	

Budgets

Budgets can be found on the Project page in the Settings area, on the Budget tab. Here is an example of what a Budget looks like on SpringAhead.

Settings / Projects / ABC Corp:Project A Active:

Overview
Details
Budget
Invoicing
People
Approvers
Tasks
Time Settings

Budget ?

Status	Description	Amount	Hours
OPEN	A budget item	6000	200

Add Budget

Purchase Orders ?

Status	Number	Amount	Hours
OPEN	123-456	120	

Add P.O. Numbers

< Details
Cancel
Save
Save and Exit
Invoicing >

List

URI Format	<pre> /project/{projectid}/budget /project/{projectid}/budget/page/{pageindex}/{pagesize}/{orderby}/{ascending} /project/{projectid}/budgets /project/{projectid}/budgets/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /projects/{projectid}/budget /projects/{projectid}/budget/page/{pageindex}/{pagesize}/{orderby}/ {ascending} /projects/{projectid}/budgets /projects/{projectid}/budgets/page/{pageindex}/{pagesize}/{orderby}/ {ascending} </pre> <p> pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </p>
Verb	GET

Access	Admin, ApiAdmin
Description	Return a list of all the budgets associated with the project specified.
Example Response	
Known issues	

Details

URI Format	<pre> /project/{projectid}/budget/{id} /project/{projectid}/budgets/{id} /projects/{projectid}/budget/{id} /projects/{projectid}/budgets/{id} </pre>
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific budget associated with the specified Project.
Example Response	
Known issues	

Edit

URI Format	<pre> /project/{projectid}/budget/{id} /project/{projectid}/budgets/{id} /projects/{projectid}/budget/{id} /projects/{projectid}/budgets/{id} </pre>
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an budget.
Example Request	
Known issues	

MultiEdit

URI Format	<pre> /project/{projectid}/budget /project/{projectid}/budgets /projects/{projectid}/budget /projects/{projectid}/budgets </pre>
Verb	POST

Description	Update specific settings on a budget.
Example Request	
Known issues	

Create

URI Format	/project/{projectid}/budget /project/{projectid}/budgets /projects/{projectid}/budget /projects/{projectid}/budgets
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new budget entries. This API supports multiple budgets in the Request. So you don't have to call over and over again to create new ones.
Example Request	
Known issues	

Delete

URI Format	/project/{projectid}/budget/{id} /project/{projectid}/budgets/{id} /projects/{projectid}/budget/{id} /projects/{projectid}/budgets/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the budget specified by the id.
Example Request	
Known issues	

MultiDelete

URI Format	/project/{projectid}/budget /project/{projectid}/budgets /projects/{projectid}/budget /projects/{projectid}/budgets
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Budgets specified by the list of ids in the request XML.

Example Request	
Known issues	

POs

POs (Purchase Orders) can be found on the Project page in the Settings area, on the Budget tab. Here is an example of what a PO looks like on SpringAhead.

Settings / Projects / ABC Corp:Project A Active:

Overview
Details
Budget
Invoicing
People
Approvers
Tasks
Time Settings

Budget ?

Status	Description	Amount	Hours
OPEN	A budget item	6000	200

Add Budget

Purchase Orders ?

Status	Number	Amount	Hours
OPEN	123-456	120	

Add P.O. Numbers

< Details
Cancel
Save
Save and Exit
Invoicing >

List

URI Format	<pre> /project/{projectid}/po /project/{projectid}/po/page/{pageindex}/{pagesize}/{orderby}/{ascending} /project/{projectid}/pos /project/{projectid}/pos/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/po /projects/{projectid}/po/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/pos /projects/{projectid}/pos/page/{pageindex}/{pagesize}/{orderby}/{ascending} </pre> <p> pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </p>
Verb	GET
Access	Admin, ApiAdmin

Description	Return a list of all the POs associated with the project specified.
Example Response	
Known issues	

Details

URI Format	/project/{projectid}/po/{id} /project/{projectid}/pos/{id} /projects/{projectid}/po/{id} /projects/{projectid}/pos/{id}
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific POs associated with the specified Project.
Example Response	
Known issues	

Edit

URI Format	/project/{projectid}/po/{id} /project/{projectid}/pos/{id} /projects/{projectid}/po/{id} /projects/{projectid}/pos/{id}
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an PO.
Example Request	
Known issues	

MultiEdit

URI Format	/project/{projectid}/po /project/{projectid}/pos /projects/{projectid}/po /projects/{projectid}/pos
Verb	POST
Description	Update specific settings on a PO.

Example Request	
Known issues	

Create

URI Format	/project/{projectid}/po /project/{projectid}/pos /projects/{projectid}/po /projects/{projectid}/pos
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new PO entries. This API supports multiple POs in the Request. So you don't have to call over and over again to create new ones.
Example Request	
Known issues	

Delete

URI Format	/project/{projectid}/po/{id} /project/{projectid}/pos/{id} /projects/{projectid}/po/{id} /projects/{projectid}/pos/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the PO specified by the id.
Example Request	
Known issues	

MultiDelete

URI Format	/project/{projectid}/po /project/{projectid}/pos /projects/{projectid}/po /projects/{projectid}/pos
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the POs specified by the list of ids in the request XML.
Example Request	

Known issues	
--------------	--

Tasks

Tasks can be found on the Project page in the Settings area, on the Tasks tab. Here is an example of what a Task looks like on SpringAhead.

Settings / Projects / ABC Corp:Project A Active:

Overview Details Budget Invoicing People Approvers **Tasks** Time Settings

Project Task

This project doesn't use tasks
 This project uses tasks and they are optional
 All tasks apply to this project
 Only selected tasks apply to this project

Tasks

Tasks	Bill Rate	Pay rate
1099 Hours	123	99

Add Tasks

< Approvers Cancel Save Save and Exit Time Settings >

List

URI Format	<pre> /project/{projectid}/task /project/{projectid}/task/page/{pageindex}/{pagesize}/{orderby}/{ascending} /project/{projectid}/tasks /project/{projectid}/tasks/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/task /projects/{projectid}/task/page/{pageindex}/{pagesize}/{orderby}/{ascending} /projects/{projectid}/tasks /projects/{projectid}/tasks/page/{pageindex}/{pagesize}/{orderby}/{ascending} pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </pre>
Verb	GET
Access	Admin, ApiAdmin

Description	Return a list of all the Tasks associated with the project specified.
Example Response	
Known issues	

Details

URI Format	/project/{projectid}/task/{id} /project/{projectid}/tasks/{id} /projects/{projectid}/task/{id} /projects/{projectid}/tasks/{id}
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific Tasks associated with the specified Project.
Example Response	
Known issues	

Edit

URI Format	/project/{projectid}/task/{id} /project/{projectid}/tasks/{id} /projects/{projectid}/task/{id} /projects/{projectid}/tasks/{id}
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an Task.
Example Request	
Known issues	

MultiEdit

URI Format	/project/{projectid}/task /project/{projectid}/tasks /projects/{projectid}/task /projects/{projectid}/tasks
Verb	POST
Description	Update specific settings on a Task.

Example Request	
Known issues	

Create

URI Format	/project/{projectid}/task /project/{projectid}/tasks /projects/{projectid}/task /projects/{projectid}/tasks
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Task entries. This API supports multiple Tasks in the Request. So you don't have to call over and over again to create new ones.
Example Request	
Known issues	

Delete

URI Format	/project/{projectid}/task/{id} /project/{projectid}/tasks/{id} /projects/{projectid}/task/{id} /projects/{projectid}/tasks/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Task specified by the id.
Example Request	
Known issues	

MultiDelete

URI Format	/project/{projectid}/task /project/{projectid}/tasks /projects/{projectid}/task /projects/{projectid}/tasks
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Task specified by the list of ids in the request XML.
Example Request	

Known issues	
--------------	--

Secondaries for Users


Rates

Rates can be found on the User page in the Settings area. Here is an example of what some Rates look like on SpringAhead. Alan Adler has a bill rate of \$75 and a pay rate of \$50.

Settings / People / Adler, Alan Active: Yes

Overview
Personal Info
Permissions
Project Rates
Mapping
Approvers

Pay Cycle This person is paid On Demand (All)

Active Dates	Bill Rate	Pay Rate	
6/1/2009 - Current	\$75.00	\$50.00	

Add Bill and Pay Rates

Overtime Rules Never pay this person overtime

List

URI Format	<pre> /user/{userid}/rate /user/{userid}/rate/page/{pageindex}/{pagesize}/{orderby}/{ascending} /user/{userid}/rates /user/{userid}/rates/page/{pageindex}/{pagesize}/{orderby}/{ascending} /users/{userid}/rate /users/{userid}/rate/page/{pageindex}/{pagesize}/{orderby}/{ascending} /users/{userid}/rates /users/{userid}/rates/page/{pageindex}/{pagesize}/{orderby}/{ascending} </pre> <p> pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </p>
Verb	GET
Access	Admin, ApiAdmin
Description	Return a list of all the Rates associated with the project specified.

Example Response	
Known issues	

Details

URI Format	<pre> /user/{userid}/rate/{id} /user/{userid}/rates/{id} /users/{userid}/rate/{id} /users/{userid}/rates/{id} </pre>
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific Rate associated with the specified User.
Example Response	
Known issues	

Edit

URI Format	<pre> /user/{userid}/rate/{id} /user/{userid}/rates/{id} /users/{userid}/rate/{id} /users/{userid}/rates/{id} </pre>
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an Rate.
Example Request	<p>Example Xml for /project/1/rate/31, set active to true</p> <pre> <SpringAhead> <UserRate> <Active>true</Active> <BillRate>120</BillRate> </UserRate> </SpringAhead> </pre>
Known issues	

MultiEdit

URI Format	<pre> /user/{userid}/rate /user/{userid}/rates /users/{userid}/rate /users/{userid}/rates </pre>
Verb	POST

Description	Update specific settings on an Rate.
Example Request	<p>Example Rate for 31 and 32, set active to true and BillRate on both</p> <pre> <SpringAhead> <UserRate> <Id>31</Id> <Active>>true</Active> <BillRate>110</BillRate> </UserRate> <UserRate> <Id>32</Id> <Active>>true</Active> <BillRate>120</BillRate> </UserRate> </SpringAhead> </pre>
Known issues	

Create

URI Format	<pre> /user/{userid}/rate /user/{userid}/rates /users/{userid}/rate /users/{userid}/rates </pre>
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Rate entries. This API supports multiple Rates in the Request. So you don't have to call over and over again to create new ones.
Example Request	<pre> <?xml version="1.0" encoding="utf-8" ?> <SpringAhead xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <UserRate> <Active>true</Active> <DateStart>2011-1-1T00:00:00</DateStart> <DateStop>2011-12-31T00:00:00</DateStop> <BillRate>120</BillRate> </UserRate> <UserRate> <Active>true</Active> <DateStart>2012-1-1T00:00:00</DateStart> <BillRate>140</BillRate> </UserRate> </SpringAhead> </pre> <p>Sending this XML will cause 2 new Rates to be created. Note that rates can not overlap. The API will not allow this to occur.</p>
Known issues	

Delete

URI Format	/user/{userid}/rate/{id} /user/{userid}/rates/{id} /users/{userid}/rate/{id} /users/{userid}/rates/{id}
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the rate specified by the id.
Known issues	

MultiDelete

URI Format	/user/{userid}/rate /user/{userid}/rates /users/{userid}/rate /users/{userid}/rates
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Rates specified by the list of ids in the request XML.
Example Request	<pre> <SpringAhead> <UserRate> <Id>31</Id> </UserRate> <UserRate> <Id>32</Id> </UserRate> </SpringAhead> </pre>
Known issues	

Reps

Reps can be found on the Project page in the Settings area, on the Approvers tab. Here is an example of what some Reps look like on SpringAhead.

Settings / Projects / ABC Corp:Project A Active: Yes

Overview Details Budget Invoicing People **Approvers** Tasks Time Settings


Manager Manager Manager approves time

Time Approvers Any approver can approve time All approvers must approve time

Only use time approvers on this page

Project Burden When calculating commissions, add a % burden to this project's pay rate

Reps

Name	Active Dates	Rep Type	Commission
Milbourne, Mark	3/5/2012 - Current	Account Manager	100000 

< People Cancel Save Save and Exit Tasks >

List	
URI Format	<pre> /user/{userid}/rep /user/{userid}/rep/page/{pageindex}/{pagesize}/{orderby}/{ascending} /user/{userid}/reps /user/{userid}/reps/page/{pageindex}/{pagesize}/{orderby}/{ascending} /users/{userid}/rep /users/{userid}/rep/page/{pageindex}/{pagesize}/{orderby}/{ascending} /users/{userid}/reps /users/{userid}/reps/page/{pageindex}/{pagesize}/{orderby}/{ascending} </pre> <p> pageindex - int pagesize - int orderby - xml element name to orderby, only string types are supported ascending - bool search - the value being searched for. </p>
Verb	GET

Access	Admin, ApiAdmin
Description	Return a list of all the Reps associated with the User specified.
Example Response	
Known issues	

Details

URI Format	/user/{userid}/rep/{id} /user/{userid}/reps/{id} /users/{userid}/rep/{id} /users/{userid}/reps/{id}
Verb	GET
Access	Admin, ApiAdmin
Description	Return details of a specific Rep associated with the specified Project.
Example Response	
Known issues	

Edit

URI Format	/user/{userid}/rep/{id} /user/{userid}/reps/{id} /users/{userid}/rep/{id} /users/{userid}/reps/{id}
Verb	POST
Access	Admin, ApiAdmin
Description	Update specific settings on an Rep.
Example Request	
Known issues	

MultiEdit

URI Format	/user/{userid}/rep /user/{userid}/reps /users/{userid}/rep /users/{userid}/reps
Verb	POST

Description	Update specific settings on a Rep.
Example Request	
Known issues	

Create

URI Format	<pre> /user/{userid}/rep /user/{userid}/reps /users/{userid}/rep /users/{userid}/reps </pre>
Verb	PUT
Access	Admin, ApiAdmin
Description	Create new Rep entries. This API supports multiple Reps in the Request. So you don't have to call over and over again to create new ones.
Example Request	
Known issues	

Delete

URI Format	<pre> /user/{userid}/rep/{id} /user/{userid}/reps/{id} /users/{userid}/rep/{id} /users/{userid}/reps/{id} </pre>
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the rep specified by the id.
Example Request	
Known issues	

MultiDelete

URI Format	<pre> /user/{userid}/rep /user/{userid}/reps /users/{userid}/rep /users/{userid}/reps </pre>
Verb	DELETE
Access	Admin, ApiAdmin
Description	Delete the Reps specified by the list of ids in the request XML.

Example Request	
Known issues	

References

REST: http://en.wikipedia.org/wiki/Representational_State_Transfer

SSL: http://en.wikipedia.org/wiki/Transport_Layer_Security

Basic Authentication: http://en.wikipedia.org/wiki/Basic_authentication

Fiddler2: <http://www.fiddler2.com/fiddler2/>

Firefox: <http://www.mozilla.com/en-US/firefox/fx/>

HTTP Server Return Codes: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>