



# API Documentation

For: **PROWORKFLOW 'PROFESSIONAL' & PROWORKFLOW 'ADVANCED'**

## INTRODUCTION:

---

This is the third release of the ProWorkflow API and consists of the components as outlined below.

An API key is available in the Client Area of the ProWorkflow website, [http://www.proworkflow.com/clients/dsp\\_login.cfm](http://www.proworkflow.com/clients/dsp_login.cfm).

For beta testing we recommend that you contact us to set up a development version of your live subscription account, [development@proworkflow.com](mailto:development@proworkflow.com).

Required version of ProWorkflow for API key acceptance is v.6.8

## COMPONENTS OF THE API:

---

- Client Area
- API Administration
- Customer Authentication
- Projects Manager API
- Tasks Manager API
- Contacts Manager API
- Time Manager API
- Upcoming Items API

## Client Area

---

An option has been built in the Client Area section of the ProWorkflow website to request an API key for the subscription account.

Once generated the “CUSTOMER KEY”, i.e. 84DC6CF8-BF78-D79D-B25B-0C0D20229F17 is stored:

- In the customer’s PWF account directory e.g.  
[https://proworkflow2.net/pwfaccountname/api/storage/customer\\_key.txt](https://proworkflow2.net/pwfaccountname/api/storage/customer_key.txt)
- In the Client Area of the PWF website, i.e. Client Area > API
- In the Admin area of the PWF subscription account, i.e. ADMINISTRATION > API Settings

## API Administration

---

**Settings Page** - This displays the Customer Key to the user but will not be editable. There is an option to restrict the IP range or allow all users; the default is set to be accessible to all the users. On every API call the IP will be checked and then access will be granted.

**History** – This displays all the API calls with the date and time stamp, the IP address, the web service name, request data and response data. A filter is provided to search for specific API calls.

## Customer Authentication

---

API users must pass the customer key in the URL string for authentication, e.g.

<https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17>. This process is done on every API call so that security issues are not violated. This component of the API is called locally from the API Caller Interface and the users don’t have to call the Authenticate web service explicitly.

## Login API

---

The Login API authenticates the user and sends the UserID with the permissions for the user. It also sends the Customer Key for any other API calls.

### Login – Authenticates the User

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/index.cfm?api\\_call=login&Username=xxx&Password=xxx](https://proworkflow2.net/pwfaccountname/api/index.cfm?api_call=login&Username=xxx&Password=xxx)
2. Method - GET
3. URL Parameters – Username and Password are mandatory.
4. Return Data – A XML data packet of UserDetails is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<Login>
  <LoginDetails>
    <Status>Success</Status>
    <Description/>
  </LoginDetails>
  <UserDetails>
    <UserID>329</UserID>
    <UserName>Julian Stone</UserName>
    <ContactTypeID>3</ContactTypeID>
    <SuperUser>1</SuperUser>
    <ShowInfo>1</ShowInfo>
    <Permissions>
      <JobsManager>
        <View>1</View>
        <Add>1</Add>
        <Edit>1</Edit>
        <Delete>1</Delete>
      </JobsManager>
      <TasksManager>
        <View>1</View>
        <Add>1</Add>
        <Edit>1</Edit>
        <Delete>1</Delete>
      </TasksManager>
      <ContactsManager>
        <View>1</View>
        <Add>1</Add>
        <Edit>1</Edit>
        <Delete>1</Delete>
      </ContactsManager>
      <TimeManager>
        <View>1</View>
        <Add>1</Add>
      </TimeManager>
    </Permissions>
  </UserDetails>
</Login>
```

```
                <Edit>1</Edit>
                <Delete>1</Delete>
            </TimeManager>
        </Permissions>
    </UserDetails>
    <CustomerKey>84DC6CF8-BF78-D79D-B25B-0C0D20229F17</CustomerKey>
</Login>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
<Login>
    <LoginDetails>
        <Status>Failure</Status>
        <Description> The user is tagged with type Other. This contact type does not have login
        permission. </Description>
    </LoginDetails>
</ Login >
```

OR

```
(This is for Advanced Account)
<?xml version="1.0" encoding="UTF-8"?>
<Login>
    <LoginDetails>
        <Status>Failure</Status>
        <Description> Staff User account not assigned to any active Team or Group.</Description>
    </LoginDetails>
</ Login >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
<Login>
    <LoginDetails>
        <Status>Failure</Status>
        <Description> Invalid username and password.</Description>
    </LoginDetails>
</ Login >
```

## Projects Manager API

---

The Project Manager API has many components to it. Below is the list of web services with examples.

### **GetAllJobs** - Retrieves all the jobs

7. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=alljobs&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=alljobs&UserID=xxx)
8. Method - GET
9. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
10. Return Data – A XML data packet of AllJobs is constructed by the web service and sent to the user.
11. Status messages include: OK or Error
12. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllJobs>
  <Status>OK</Status>
  <Job_Count>1</Job_Count>
  <Jobs>
    <Job>
      <id>1</id>
      <title> Big Company</title>
      <client>Big Company</client>
      <number>BC0001</number>
    </Job>
  </Jobs>
</AllJobs>
```

### **GetAllActiveJobs** - Retrieves all the active jobs

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=activejobs&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=activejobs&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data – A XML data packet of AllActiveJobs is constructed by the web service and sent to the user.
5. Status messages include: OK or Error.

## 6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllActiveJobs>
  <Status>OK</Status>
  <Job_Count>1</Job_Count>
  <Jobs>
    <Job>
      <id>1</id>
      <title>(BC) Big Company email </title>
      <client>Big Company</client>
      <number>BC0001</number>
    </Job>
  </Jobs>
</ AllActiveJobs>
```

## **GetAllOverDueJobs** - Retrieves all the overdue jobs

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=overduejobs&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=overduejobs&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data – A XML data packet of AllOverDueJobs is constructed by the web service and sent to the user.
5. Status messages include: OK or Error.
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllOverDueJobs>
  <Status>OK</Status>
  <Job_Count>1</Job_Count>
  <Jobs>
    <Job>
      <id>1</id>
      <title>Email Campaign</title>
      <client>Big Company</client>
      <number>BC0001</number>
    </Job>
  </Jobs>
</ AllOverDueJobs>
```

## GetAllArchivedJobs - Retrieves all the archived jobs

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=archivedjobs&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=archivedjobs&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data – A XML data packet of AllArchivedJobs is constructed by the web service and sent to the user.
5. Status messages include: OK or Error.
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllArchivedJobs>
  <Status>OK</Status>
  <Job_Count>1</Job_Count>
  <Jobs>
    <Job>
      <id>1</id>
      <title>Email Campaign </title>
      <client>Big Company</client>
      <number>BC0001</number>
    </Job>
  </Jobs>
</ AllArchivedJobs >
```

## GetSpecificJob - Gets the data of a specific project

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=specificjob&JobId=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=specificjob&JobId=xxx)
2. Method - GET
3. URL Parameters – JobId is mandatory.
4. Return Data – A XML data packet of a SpecificJob detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error.

## 6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< SpecificJob>
  <Status>OK</Status>
  <Job_Count>1</Job_Count>
  <Job>
    <id>1</id>
    <title>Email Campaign</title>
    <client>Big Company</client>
    <number>BC0001</number>
    <Startdate>01-Jun-2009</Startdate>
    <Duedate>27-Jun-2009</Duedate>
    <Completeddate>27-Jun-2009</Completeddate>
    <Invoiced>N</Invoiced>
    <Paid>N</Paid>
    <AccountedFor>N</AccountedFor>
    <AllocatedTime>426</AllocatedTime>
    <TotalTimeSpent></TotalTimeSpent>
    <JobComplete>0</JobComplete>
    <JobDescription>Time limited project for our favorite clients!</JobDescription>
    <Notes> </Notes>
    <JobStatus>14</JobStatus>
    <JobAssignedTo>x5x</JobAssignedTo>
    <JobAllocatedhours>7</JobAllocatedhours>
    <JobAllocatedminutes>6</JobAllocatedminutes>
    < Quotedprice> 5234.625</ Quotedprice>
    <ClientLogin>1</ClientLogin>
    <InvoicedPrice>798.75</InvoicedPrice>
    <JobCreatedBy>5</JobCreatedBy>
    <JobCategory>Production</JobCategory>
    <JobManager>5</JobManager>
    <JobManagerDetails>
      <ContactId>5</ContactId>
      <ContactCategoryID>1</ContactCategoryID>
      <ContactTypeID>3</ContactTypeID>
      <CompanyId>Big Company</CompanyId>
      <FullName>Aubrey Apple</FullName>
      <AddressLine1>328 Two Chain Road</AddressLine1>
      <AddressLine2>Rolleston</AddressLine2>
      <AddressLine3> </AddressLine3>
      <WorkPhone>+643 347 3221</WorkPhone>
      <MobilePhone>+6421 347 3223</MobilePhone>
      <EmailAddress/>
      <City>Christchurch</City>
      <State/>
      <ZipCode>7675</ZipCode>
      <Country> New Zealand</Country>
    </JobManagerDetails>
  </Job>
</ SpecificJob>
```

```
</JobManagerDetails>

  </Job>
</ SpecificJob>
```

**DeleteJob** - Delete job, insert in the audit log and sends an email alert.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=deletejob&JobId=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=deletejob&JobId=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – JobId and UserId are mandatory.
4. Note - User Permissions are also checked for while deleting the jobs.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error.
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<DeleteJob>
  <Status>Success</Status>
  <Details>
    <Message>Job deleted successfully</Message>
  </Details>
</DeleteJob>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
<DeleteJob>
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action. </Message>
  </Details>
</DeleteJob>
```

**AddJob**- Adds a job, inserts in the audit log and sends an email alert.

1. Web service call - e.g.

[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addjob&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addjob&UserID=xxx)

2. Method – POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<NewJob>
```

```
< JobAssignedTo >5</ JobAssignedTo >
```

```
< JobCreatedBy >5</ JobCreatedBy >
```

```
< ContractorsAssigned >5</ ContractorsAssigned >
```

```
< AllocatedTime_Hour >2</ AllocatedTime_Hour >
```

```
< AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
```

```
< StartDate>02-Jan-2009</ StartDate>
```

```
< DueDate>02-Feb-2009</ DueDate>
```

```
< CompDate>31-Jan-2009</ CompDate>
```

```
< JobClientID>COM34</ JobClientID> (IF Company is a client append COM in front of the id or person
```

then PER in front of the id e.g. PER)

```
< JobDescription>This job is added using the API!</ JobDescription>
```

```
< JobNumber>BC0024</ JobNumber>
```

```
< JobTitle>Website Update</ JobTitle>
```

```
< JobStatus>12</ JobStatus>
```

```
< Invoiced>N</Invoiced> (Y for Yes, N for NO)
```

```
< paid>N</paid> (Y for Yes, N for NO)
```

```
< AccountedFor>N</AccountedFor> (Y for Yes, N for NO)
```

```
< ClientLogin>1</ClientLogin> (1 for Yes, 0 for No)
```

```
< QuotedPrice>135.00</QuotedPrice>
```

```
< InvoicedPrice>78</InvoicedPrice>
```

```
< notes>testing testing testing</notes>
```

```
< JobComplete>Active</JobComplete> (or Complete)
```

```
</ NewJob >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – UserID is mandatory.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message. If it is a success then a link of the job is also sent in the return data.
6. Status messages include: Success, Failure or Error.

## 7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AddJob>
  <Status>Success</Status>
  <Details>
    <Message>Click on the link to view the job.
    https://proworkflow2.net/pwfaccountname/index.cfm?Fuseaction=jobs&Fusesubaction=jobdetails&Jobs_currentJobI
    D= 1042
    </Message>
    <JobID>1042</JobID>
  </Details>
</AddJob>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
<AddJob>
  <Status>Failure</Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.</Message>
  </Details>
</AddJob>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
<AddJob>
  <Status>Failure</Status>
  <Details>
    <Message>>Not a valid client</Message>
  </Details>
</AddJob>
```

**EditJob**- Edits a job, inserts in the audit log and sends an email alert.

1. Web service call - e.g.

https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\_call=editjob&UserID=xxx

2. Method – POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<EditJob>
```

```
<JobID>945</JobID>
```

```
< JobAssignedTo >5</ JobAssignedTo >
```

```
< JobCreatedBy >5</ JobCreatedBy >
```

```
< ContractorsAssigned >29</ ContractorsAssigned >
```

```
< AllocatedTime_Hour >2</ AllocatedTime_Hour >
```

```
< AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
```

```
< StartDate>02-Jan-2009</ StartDate>
```

```
< DueDate>02-Feb-2009</ DueDate>
```

```
< CompDate>31-Jan-2009</ CompDate>
```

```
< JobClientID>COM34</ JobClientID> (IF Company as a client append COM in front of the id or person
```

then PER in front of the id e.g. PER16)

```
< JobDescription>This job is added using the API!</ JobDescription>
```

```
< JobNumber>Test</ JobNumber>
```

```
< JobTitle>API Testing</ JobTitle>
```

```
< JobStatus>12</ JobStatus>
```

```
< Invoiced>N</Invoiced> (Y for Yes, N for NO)
```

```
< paid>N</paid>(Y for Yes, N for NO)
```

```
< AccountedFor>N</AccountedFor> (Y for Yes, N for NO)
```

```
< ClientLogin>1</ClientLogin> (1 for Yes, 0 for No)
```

```
< QuotedPrice>135.00</QuotedPrice>
```

```
< InvoicedPrice>78</InvoicedPrice>
```

```
< notes>testing testing testing</notes>
```

```
< JobComplete>Active</JobComplete> (or Complete)
```

```
</ EditJob >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – UserID is mandatory.

5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message. If it is a success then a link of the job is also sent in the return data.

6. Status messages include: Success, Failure or Error.

## 7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<EditJob>
  <Status>Success</Status>
  <Details>
    <Message>Click on the link to view the job.
    https://proworkflow2.net/pwfaccountname/index.cfm?Fuseaction=jobs&Fusesubaction=jobdetails&Jobs_c
    urrentJobID= 1042
    </Message>
    <JobID>1042</JobID>
  </Details>
</ EditJob >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditJob >
  <Status>Failure</Status>
  <Details>
    <Message>Error: You don't have permission to perform this action. </Message>
  </Details>
</ EditJob >
```

## Tasks Manager API

---

The Tasks Manager API has many components to it. Below is the list of web services with examples.

### GetAllTasks - Retrieves all the Tasks

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=alltasks&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=alltasks&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data – A XML data packet of AllTasks is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllTasks>
  <Status>OK</Status>
  <Task_Count>1</Task_Count>
  <Tasks>
    <Task>
      <taskid>7911</taskid>
      <jobid>37</jobid> (if a jobtask else if a General Task it is an empty field)
      <title>Test Task</title>
      <taskstatus>A </taskstatus>
      <datetostart>11-Feb-2009</datetostart>
      <datedue>15-Feb-2009-02</datedue>
      <taskcreatedby>5</taskcreatedby>
      <taskassignedto>x5</taskassignedto>
    </Task>
  </Tasks>
</AllTasks>
```

### GetAllActiveJobTasks - Retrieves all the active Job Tasks

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=activetasks&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=activetasks&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”

4. Return Data – A XML data packet of AllActiveTasks is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllActiveTasks >
  <Status>OK</Status>
  <Task_Count>1</Task_Count>
  <Tasks>
    <Task>
      <taskid>7911</taskid>
      <jobid>537</jobid> (if a jobtask else it is an empty field)
      <title>Test Task</title>
      <taskstatus>A </taskstatus>
      <datetostart>11-Feb-2009</datetostart>
      <datedue>15-Feb-2009</datedue>
      <taskcreatedby>5</taskcreatedby>
      <taskassignedto>x5x</taskassignedto>
    </Task>
  </Tasks>
</ AllActiveTasks >
```

#### **GetAllOverDueTasks** - Retrieves all the overdue Job Tasks

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=overduetasks&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=overduetasks&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data - A XML data packet of AllOverDueTasks is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllOverDueTasks >
  <Status>OK</Status>
  <Task_Count>1</Task_Count>
  <Tasks>
    <Task>
      <taskid>7911</taskid>
      <jobid>537</jobid> (if a jobtask else it is an empty field)
      <title>Test Task</title>
      <taskstatus>A </taskstatus>
```

```

        <datetostart>11-Feb-2009</datetostart>
        <datedue>15-Feb-2009</datedue>
        <taskcreatedby>5</taskcreatedby>
        <taskassignedto>x5x</taskassignedto>
    </Task>
</Tasks>
</ AllOverDueTasks >

```

### GetAllArchivedTasks - Retrieves all the archived Job Tasks

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=archivedtasks&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=archivedtasks&UserID=xxx)
2. Method - GET
3. URL Parameters – UserID is not mandatory. If no id is provided then “SHOW INFOR FOR USER = ALL STAFF USERS”
4. Return Data – A XML data packet of AllArchivedTasks is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```

<?xml version="1.0" encoding="UTF-8"?>
< AllArchivedTasks >
    <Status>OK</Status>
    <Task_Count>1</Task_Count>
    <Tasks>
        <Task>
            <taskid>7911</taskid>
            <jobid>537</jobid> (if a jobtask else it is an empty field)
            <title>Test Task</title>
            <taskstatus>A </taskstatus>
            <datetostart>11-Feb-2009</datetostart>
            <datedue>15-Feb-2009</datedue>
            <taskcreatedby>5</taskcreatedby>
            <taskassignedto>x5x</taskassignedto>
        </Task>
    </Tasks>
</ AllArchivedTasks >

```

## GetSpecificTask - Gets the data of a specific Task

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=specifictask&TaskId=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=specifictask&TaskId=xxx)
2. Method - GET
3. URL Parameters – TaskId is mandatory.
4. Return Data – A XML data packet of a SpecificTask detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< GetSpecificTask>
  <Status>OK</Status>
  <Task_Count>1</Task_Count>
  < Task>
    <taskid>7911</taskid>
    <jobid>537</jobid>
    <title>Test Task</title>
    <taskstatus>A </taskstatus>
    <datetostart>11-Feb-2009</datetostart>
    <datedue>15-Feb-2009</datedue>
    <taskcreatedby>5</taskcreatedby>
    <taskassignedto> x53x1x </taskassignedto>
    < Taskpriority>3</ Taskpriority>
    < allocatedtime>0</allocatedtime>
    < deleted>0</deleted>
    < compdate> </compdate>
    < taskorder >9.0</ taskorder >
    <taskbilling>1</taskbilling>
    <taskdescription></taskdescription>
    <allocatedtime_hours>3</allocatedtime_hours>
    <allocatedtime_minutes>0</allocatedtime_minutes>
    <totaltimespent>3:12</totaltimespent>
    <taskstyle_bh>1</taskstyle_bh>
    <taskstyle_b>0</taskstyle_b>
    <tracking>1</tracking >
    <trackerid>157</trackerid >{this data is only provided if the tracking field is set to 1}
  </Task>
</ GetSpecificTask>
```

## GetSpecificJobTask - Gets the data of a specific JobTask

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=specificjobtasks&JobID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=specificjobtasks&JobID=xxx)
2. Method - GET
3. URL Parameters – JobID is mandatory.
4. Return Data – A XML data packet of AllJobTasks detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllJobTasks>
  <Status>OK</Status>
  <Job>
    <JobID>537</JobID>
    <JobTitle>Test Project</JobTitle>
    <JobNumber>BC0005</JobNumber>
    <Tasks>
      <Task>
        <taskid>7911</taskid>
        <jobid>537</jobid>
        <title>Test Task</title>
        <taskstatus>A </taskstatus>
        <datetostart>11-Feb-2009</datetostart>
        <datedue>15-Feb-2009</datedue>
        <taskcreatedby>5</taskcreatedby>
        <taskassignedto> x53x1x </taskassignedto>
        <Taskpriority>3</Taskpriority>
        <allocatedtime>0/ allocatedtime>
        <deleted>0</deleted>
        <comdate></comdate>
        <taskorder >9.0</taskorder >
        <taskbilling>1</taskbilling>
        <taskdescription></taskdescription>
        <allocatedtime_hours>3</allocatedtime_hours>
        <allocatedtime_minutes>0</allocatedtime_minutes>
        <totaltimespent>3:12</totaltimespent>
        <taskstyle_bh>1</taskstyle_bh>
        <taskstyle_b>0</taskstyle_b>
        <tracking>1</tracking >
        <trackerid>157</ trackerid >{this data is only provided if the tracking field is
          set to 1}
      </Task>
    </Tasks>
  </Job>
```

</ AllJobTasks >

**DeleteTask** - Delete task, insert in the audit log and sends an email alert.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=deletetask&TaskId=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=deletetask&TaskId=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – TaskId and UserId are mandatory.
4. Note - User Permissions are also checked for while deleting the tasks.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteTask>
    <Status>Success</Status>
    <Details>
        <Message>Task deleted</Message>
    </Details>
</ DeleteTask>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteTask>
    <Status> Failure </Status>
    <Details>
        <Message>Error: You don't have permission to perform this action.</Message>
    </Details>
</ DeleteTask>
```

**GetAllTaskContacts** – Retrieves all the contact details for a particular task

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=taskcontacts&TaskId=xxx](https://proworkflow2.net/pwfaccountname/api/tasks.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=taskcontacts&TaskId=xxx)
2. Method - GET
3. URL Parameters – TaskId is mandatory
4. Return Data – A XML data packet of AllTasksContacts detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllTasksContacts>
  <Status>OK</Status>
  <Contact_Count>1</Contact_Count>
  <ContactsList>
    <Contact>
      <ContactId>5</ContactId>
      <ContactCategoryId>1</ContactCategoryId>
      <ContactTypeId>3</ContactTypeId>
      <CompanyId>5</CompanyId>
      <FullName>Aubrey Apple</FullName>
      <AddressLine1>328 Two Chain Road</AddressLine1>
      <AddressLine2>Rolleston</AddressLine2>
      <AddressLine3> </AddressLine3>
      <WorkPhone>+643 347 3221</WorkPhone>
      <MobilePhone>+6421 347 3223</MobilePhone>
      <EmailAddress/>
      <City>Christchurch</City>
      <State/>
      <ZipCode>7675</ZipCode>
      <Country> New Zealand</Country>
    </Contact>
  </ContactsList>
</AllTasksContacts>
```

**AddTask** - Adds task, insert in the audit log and sends an email alert.

1. Web service call - e.g.

[https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addtask&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addtask&UserID=xxx)

2. Method – POST

3. XML Request Data – e.g.

If Account Plan is Professional then use this XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<NewTask>
  < TaskAssignedTo>5</ TaskAssignedTo >
  < AllocatedTime_Hour >2</ AllocatedTime_Hour >
  < AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
  < DateToStart >02-Jan-2009</ DateToStart >
  < DueDate >02-Feb-2009</ DueDate >
  < TaskStyleBH >1</ TaskStyleBH >
  < TaskStyleB ></ TaskStyleB >
  < TaskTitle > Check Visuals </ TaskTitle >
  < TaskBillable >1</ TaskBillable > (1 for Yes, 0 for No)
  < NewTaskOrder >1.2</ NewTaskOrder >
  < JobID >957</ JobID > (If this is a Job Task)
  < TaskDescription >Testing</ TaskDescription >
  < TaskPriority >3</ TaskPriority >
</ NewTask >
```

Else If Account Plan is Advanced use this XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<NewTask>
  < StaffAssignedTo >5</ StaffAssignedTo >
  < ClientsAssignedTo >89</ ClientsAssignedTo >
  < ContractorsAssignedTo >43</ ContractorsAssignedTo>
  < AllocatedTime_Hour >2</ AllocatedTime_Hour >
  < AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
  < DateToStart >02-Jan-2009</ DateToStart >
  < DueDate >02-Feb-2009</ DueDate >
  < TaskStyleBH ></ TaskStyleBH >
  < TaskStyleB >1</ TaskStyleB >
  < TaskTitle >Check Visuals</ TaskTitle >
  < TaskBillable >1</ TaskBillable > (1 for Yes, 0 for No)
  < NewTaskOrder >1.2</ NewTaskOrder >
  < JobID >957</ JobID > (If this is a Job Task)
  < TaskDescription >Testing</ TaskDescription >
  < TaskPriority >3</ TaskPriority >
</ NewTask >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – UserID is mandatory.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTask>
  <Status>Success</Status>
  <Details>
    <Message>Task added</Message>
    <TaskID>1796</TaskID>
  </Details>
</ AddTask>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTask>
  <Status> Failure </Status>
  <Details>
    <Message> Error: Please define which type of task is it.</Message>
  </Details>
</ AddTask>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTask>
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ AddTask>
```

**EditTask**- Edits task, insert in the audit log and sends an email alert.

1. Web service call - e.g.  
`https://proworkflow2.net/pwfaccountname/api/projects.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=edittask&UserID=xxx`
2. Method – POST
3. XML Request Data – e.g.

If Account Plan is Professional then use this XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<EditTask>
  <TaskID>56</TaskID>
  <TaskAssignedTo>5</TaskAssignedTo >
  <AllocatedTime_Hour >2</ AllocatedTime_Hour >
  <AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
  <DateToStart >02-Jan-2009</ DateToStart >
  <DueDate >02-Feb-2009</ DueDate >
  <DateCompleted>02-Feb-2009</ DateCompleted>
  <TaskStyleBH >1</ TaskStyleBH >
  <TaskStyleB ></ TaskStyleB >
  <TaskTitle > Check Visuals </ TaskTitle >
  <TaskBillable >1</ TaskBillable > (1 for Yes, 0 for No)
  <NewTaskOrder >1.2</ NewTaskOrder >
  <JobID >957</ JobID > (If this is a Job Task)
  <TaskDescription >Testing</ TaskDescription >
  <TaskPriority >3</ TaskPriority >
  <TaskStatus>C</ TaskStatus>(A for Active , C for Completed)
</ EditTask >
```

Else If Account Plan is Advanced use this XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<EditTask >
  <TaskID>56</TaskID>
  <StaffAssignedTo >5</ StaffAssignedTo >
  <ClientsAssignedTo >89</ ClientsAssignedTo >
  <ContractorsAssignedTo >43</ ContractorsAssignedTo>
  <AllocatedTime_Hour >2</ AllocatedTime_Hour >
  <AllocatedTime_Minutes>57</ AllocatedTime_Minutes>
  <DateToStart >02-Jan-2009</ DateToStart >
  <DueDate >02-Feb-2009</ DueDate >
  <DateCompleted>02-Feb-2009</ DateCompleted>
  <TaskStyleBH >1</ TaskStyleBH >
  <TaskStyleB ></ TaskStyleB >
  <TaskTitle > Check Visuals </ TaskTitle >
  <TaskBillable >1</ TaskBillable > (1 for Yes, 0 for No)
  <NewTaskOrder >1.2</ NewTaskOrder >
```

```
< JobID >957</ JobID > (If this is a Job Task)
< TaskDescription >Testing</ TaskDescription >
< TaskPriority >3</ TaskPriority >
< TaskStatus>C</ TaskStatus>(A for Active , C for Completed)
</ EditTask >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – UserID and TaskID are mandatory .
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< EditTask>
  <Status>Success</Status>
  <Details>
    <Message>Task Edited</Message>
    <TaskID>1796</TaskID>
  </Details>
</ EditTask >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditTask >
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action. </Message>
  </Details>
</ EditTask >
```

## Contacts Manager API

---

### GetAllContacts - Gets all the contacts from PWF

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allcontacts](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allcontacts)
2. Method - GET
3. Return Data – A XML data packet of AllContacts detail is constructed by the web service and sent to the user.
4. Status messages include : OK or Error
5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllContacts>
  <Status>OK</Status>
  <Contacts_Count>1</Contacts_Count>
  <Contacts>
    <Contact>
      <ContactId>329</ContactId>
      <ContactTypeID>3</ContactTypeID>
      <CompanyId>21</CompanyId>
      <FullName>Julian Stone</FullName>
      <AddressLine1>328 Two Chain Road</AddressLine1>
      <AddressLine2> Rolleston</AddressLine2>
      <AddressLine3/>
      <WorkPhone> 123456 (Ext.124) </WorkPhone>
      <MobilePhone> +643 347 3223 </ MobilePhone >
      <EmailAddress>sales@proworkflow.com</EmailAddress>
      <City> Christchurch</City>
      <State/>
      <ZipCode>7675 <ZipCode>
      <Country> New Zealand</Country>
    </Contact>
  </Contacts>
</AllContacts>
```

### GetAllCompanies - Gets all the companies from PWF

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allcompanies](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allcompanies)
2. Method - GET
3. Return Data – A XML data packet of AllCompanies detail is constructed by the web service and sent to the user.

4. Status messages include: OK or Error

5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllCompanies >
  <Status>OK</Status>
  < Company _Count>1</ Company _Count>
  < Companies >
    <Company>
      <CompanyId>168</CompanyId>
      <ContactTypeID>3</ContactTypeID>
      <CompanyName><![CDATA[Active Medical Equipment]]></CompanyName>
      <AddressLine1>15 Lansbury Ave</AddressLine1>
      <AddressLine2>Merivale</AddressLine2>
      <AddressLine3>splat</AddressLine3>
      <WorkPhone>0211419583</WorkPhone>
      <Fax>033473221</Fax>
      <EmailAddress/>
      <City>Wellington</City>
      <State>Wellington</State>
      <ZipCode>8052</ZipCode>
      <Country>USA</Country>
    </Company>
  </ Companies >
</ AllCompanies >
```

**GetAllCompanyContacts** - Gets all the company details together with contacts details for that company from PWF

1. Web service call - e.g.

[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allcompanycontacts](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allcompanycontacts)

2. Method - GET

3. Return Data – A XML data packet of AllCompanies detail is constructed by the web service and sent to the user.

4. Status messages include: OK or Error

5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AllCompanies >
  <Status>OK</Status>
  < Company _Count>1</ Company _Count>
  < Companies >
    <Company>
      <CompanyId>168</CompanyId>
```

```

<ContactTypeID>3</ContactTypeID>
<CompanyName><![CDATA[Big Company ]]></CompanyName>
<AddressLine1>15 Lansbury Ave</AddressLine1>
<AddressLine2>Merivale</AddressLine2>
<AddressLine3>The Grange</AddressLine3>
<WorkPhone>0211419583</WorkPhone>
<Fax>033473221</Fax>
<EmailAddress/>
<City>Wellington</City>
<State>Wellington</State>
<ZipCode>8052</ZipCode>
<Country>USA</Country>
<Contacts>
  <Contacts_Count>1</Contacts_Count>
  <Contact>
    <ContactId>277</ContactId>
    <CompanyId>168</CompanyId>
    <FullName>Jessica Rabbit</FullName>
    <AddressLine1>4878 Jones Lane</AddressLine1>
    <AddressLine2/>
    <AddressLine3/>
    <WorkPhone/>
    <MobilePhone/>
    <EmailAddress>jrabbit@bigcompany.com</EmailAddress>
    <City>San Diego</City>
    <State>CA</State>
    <ZipCode>92111</ZipCode>
    <Country>USA</Country>
  </Contact>
</Contacts>
</Company>
</ Companies >
</ AllCompanies >

```

**GetSpecificContactDetails**- Gets all the details of the specified contact from PWF

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=specificcontact&ContactID=3](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=specificcontact&ContactID=3)
2. Method - GET
3. URL Parameters – ContactID is mandatory
4. Return Data – A XML data packet of ContactDetails detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error

## 6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ContactDetails>
  <Status>OK</Status>
  <Contact_Count>1</Contact_Count>
  <Details>
    <ContactId>5</ContactId>
    <ContactCategoryId>1</ContactCategoryId>
    <ContactTypeID>3</ContactTypeID>
    <CompanyId>21</CompanyId>
    <CompanyName><![CDATA[Big Company]]></CompanyName>
    <ContactJobTitle>CEO</ContactJobTitle>
    <FullName>Aubrey Apple</FullName>
    <AddressLine1>328 Two Chain Road</AddressLine1>
    <AddressLine2>Rolleston</AddressLine2>
    <AddressLine3> </AddressLine3>
    <WorkPhone>+643 347 3221</WorkPhone>
    <MobilePhone>+6421 347 3223</MobilePhone>
    <HomePhone>+6421 347 3223</HomePhone>
    <Fax>+6421 347 3225</Fax>
    <AllowLogin>1</AllowLogin>
    <LoginName>demo</LoginName>
    <Password>demo</Password>
    <PrimaryContact>0</PrimaryContact>
    <PhoneExtension>ext</PhoneExtension>
    <EmailAddress/>
    <City>Christchurch</City>
    <State/>
    <ZipCode>7675</ZipCode>
    <Country> New Zealand</Country>
  </Details>
</ContactDetails>
```

**GetSpecificCompanyDetails**- Gets all the details of the specified company together with the contacts information from PWF

1. Web service call - e.g.  
[https://proworkflow2.net/pwfacountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=specificcompany&CompanyID=3](https://proworkflow2.net/pwfacountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=specificcompany&CompanyID=3)
2. Method - GET
3. URL Parameters – CompanyID is mandatory
4. Return Data – A XML data packet of CompanyDetails detail is constructed by the web service and sent to the user.
5. Status messages include: OK or Error
6. Example of return data:

```

<?xml version="1.0" encoding="UTF-8"?>
<CompanyDetails>
  <Status>OK</Status>
  <Company_Count>1</Company_Count>
  <Details>
    <CompanyId>25</CompanyId>
    <CompanyName><![CDATA[Big Company]]></CompanyName>
    <ContactTypeID>3</ContactTypeID>
    <ContactCategoryID>1</ContactCategoryID>
    <CompanyCode>BC</CompanyCode>
    <AddressLine1>4878 Jones Lane </AddressLine1>
    <AddressLine2/>
    <AddressLine3/>
    <WorkPhone>1615 278 1451</WorkPhone>
    <Fax/>
    <EmailAddress>jrabbit@bigcompany.com</EmailAddress>
    <City>San Diego</City>
    <State>CA</State>
    <ZipCode>92111</ZipCode>
    <Country/>
    <WebsiteURL>www.proworkflow.com</WebsiteURL>
    <Notes/>
    <Contacts>
      <Contacts_Count>1</Contacts_Count>
      <Contact>
        <ContactId>321</ContactId>
        <ContactCategoryID>1</ContactCategoryID>
        <ContactTypeID>3</ContactTypeID>
        <CompanyId>25</CompanyId>
        <CompanyName><![CDATA[Big Company]]></CompanyName>
        <ContactJobTitle>CEO</ContactJobTitle>
        <FullName>Steve Austin</FullName>
        <AddressLine1>4878 Jones Lane </AddressLine1>
        <AddressLine2>Cheltenham</AddressLine2>
        <AddressLine3> </AddressLine3>
        <WorkPhone>+443 347 3221</WorkPhone>
        <MobilePhone>+4421 347 3223</MobilePhone>
        <HomePhone>+4421 347 3223</HomePhone>
        <Fax>+44 7 347 3225</Fax>
        <AllowLogin>1</AllowLogin>
        <LoginName>demo</LoginName>
        <Password>demo</Password>
        <PrimaryContact>0</PrimaryContact>
        <PhoneExtension>ext</PhoneExtension>
        <EmailAddress/>
        <City>Christchurch</City>
        <State/>
        <ZipCode>7675</ZipCode>
        <Country> New Zealand</Country>
      </Contact>
    </Contacts>
  </Details>
</CompanyDetails>

```

```
        </Contact>
    </Contacts>
</Details>
</CompanyDetails>
```

### **GetAllClientDetails-** Gets all client details from PWF

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allclients](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allclients)
2. Method - GET
3. Return Data – A XML data packet of AllClients detail is constructed by the web service and sent to the user.
4. Status messages include: OK or Error
5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllClients>
    <Status>OK</Status>
    <Client_Count>114</Client_Count>
    <Clients>
        <Client>
            <CompanyId>168</CompanyId>
            <ContactTypeID>1</ContactTypeID>
            <CompanyName><![CDATA[Other Company]]></CompanyName>
            <AddressLine1>952 Plane Court</AddressLine1>
            <AddressLine2>Unit 5</AddressLine2>
            <AddressLine3>Merivale</AddressLine3>
            <WorkPhone>61211419583</WorkPhone>
            <Fax>6133473221</Fax>
            <EmailAddress/>
            <City>Sydney</City>
            <State>NSW</State>
            <ZipCode>8052</ZipCode>
            <Country>Australia</Country>
        </Client>
    </Clients>
</AllClients>
```

### **GetAllContactCategory-** Gets all the contact categories

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allcontactcategory](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allcontactcategory)
2. Method - GET

3. Return Data – A XML data packet of AllContactCategories detail is constructed by the web service and sent to the user.
4. Status messages include: OK or Error
5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllContactCategories>
  <Status>OK</Status>
  <ContactCats_Count>14</ContactCats_Count>
  <ContactCategories>
    <Category>
      <ContactCatID>11</ContactCatID>
      <CategoryName>Business</CategoryName>
      <DisplayOrder>1</DisplayOrder>
    </Category>
  </ContactCategories >
</ AllContactCategories >
```

#### **GetAllContactTypes** - Retrieves all the contact types

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=allcontacttypes](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=allcontacttypes)
2. Method - GET
3. Return Data – A XML data packet of AllContactTypes detail is constructed by the web service and sent to the user.
4. Status messages include: OK or Error
5. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AllContactTypes>
  <Status>OK</Status>
  <ContactTypes_Count>4</ContactTypes_Count>
  <ContactTypes>
    <ContactType>
      <ContactTypeID>1</ContactTypeID>
      <ContactType>Client</ContactType>
    </ContactType>
    <ContactType>
      <ContactTypeID>2</ContactTypeID>
      <ContactType>Contractor</ContactType>
    </ContactType>
    <ContactType>
      <ContactTypeID>3</ContactTypeID>
      <ContactType>Internal Staff</ContactType>
    </ContactType>
  </ContactTypes>
```

```
<ContactType>
  <ContactTypeID>4</ContactTypeID>
  <ContactType>Other</ContactType>
</ContactType>
</ContactTypes>
</AllContactTypes>
```

**AddContact**- Adds a contact to the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addcontact](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addcontact)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< NewContact >
  < UserID>329</ UserID >
  < ContactTypeID>3</ ContactTypeID>
  < ContactCatID>1</ ContactCatID>
  < ContactJobTitle >CEO</ ContactJobTitle >
  < ContactFirstName >Julian</ ContactFirstName >
  < ContactLastName >Stone</ ContactLastName >
  < HomePhone > +643 347 3221</ HomePhone >
  < WorkPhone >+643 347 3222</ WorkPhone >
  < MobilePhone >+643 347 3223</ MobilePhone >
  < ContactFax>+643 347 3223</ ContactFax>
  < HomeEmail >julian@proworkflow.com</ HomeEmail >
  < AllowLogin >1</ AllowLogin > (1 for Yes, 0 for No)
  < LoginName >julian</ LoginName >
  < Password >stone</ Password >
  < companyID >6</ companyID >
  < PrimaryContact > 1</ PrimaryContact > (1 for Yes, 0 for No)
  < PhoneExt >124</ PhoneExt >
  < ContactAddress1>328 Two Chain Road</ ContactAddress1>
  < ContactAddress2>Rolleston</ ContactAddress2>
  < ContactAddress3> </ ContactAddress3>
  < ContactCity>Christchurch</ ContactCity>
  < ContactState></ ContactState>
  < ContactZipCode>7675</ ContactZipCode>
  < ContactCountry> New Zealand</ ContactCountry>
</ NewContact>
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – None
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContact>
  <Status>Success</Status>
```

```
        <Details>
            <Message>Contact added</Message>
            <ContactID>6</ ContactID >
        </Details>
    </ AddContact>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContact >
    <Status> Failure </Status>
    <Details>
        <Message>You have reached the user limit. Please contact PWF support to upgrade. </Message>
    </Details>
</ AddContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContact >
    <Status> Failure </Status>
    <Details>
        <Message>Note: only Staff, Clients and Contractors may have login access to ProWorkflow. Please
        set Allow Login to "NO" or change your contact type. </Message>
    </Details>
</ AddContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContact >
    <Status> Failure </Status>
    <Details>
        <Message>The Log-in Name and Password you entered are already taken. Please select an
        alternative combination. </Message>
    </Details>
</ AddContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContact >
    <Status> Failure </Status>
    <Details>
        <Message>Error: You don't have permission to perform this action.. </Message>
    </Details>
</ AddContact >
```

## EditContact- Edit a contact in the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=editcontact](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=editcontact)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< EditContact >
  < UserID>5</ UserID >
  < ContactID>329</ ContactID>
  < ContactTypeID>3</ ContactTypeID>
  < ContactCatID>1</ ContactCatID>
  < ContactJobTitle >CEO</ ContactJobTitle >
  < ContactFirstName >Julian</ ContactFirstName >
  < ContactLastName >Stone</ ContactLastName >
  < HomePhone >+643 347 3221</ HomePhone >
  < WorkPhone >+643 347 3222</ WorkPhone >
  < MobilePhone >+643 347 3223</ MobilePhone >
  < ContactFax>+643 347 3223</ ContactFax>
  < HomeEmail >julian@proworkflow.com</ HomeEmail >
  < AllowLogin >1</ AllowLogin > (1 for Yes, 0 for No)
  < LoginName >julian</ LoginName >
  < Password >stone</ Password >
  < companyID >6</ companyID >
  < PrimaryContact > 1</ PrimaryContact > (1 for Yes, 0 for No)
  < PhoneExt >124</ PhoneExt >
  < ContactAddress1>328 Two Chain Road</ ContactAddress1>
  < ContactAddress2>Rolleston</ ContactAddress2>
  < ContactAddress3> </ ContactAddress3>
  < ContactCity>Christchurch</ ContactCity>
  < ContactState></ ContactState>
  < ContactZipCode>7675</ ContactZipCode>
  < ContactCountry> New Zealand</ ContactCountry>
</ EditContact>
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – None
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< EditContact>
```

```
<Status>Success</Status>
<Details>
  <Message>Contact edited</Message>
  <ContactID>6</ ContactID >
</Details>
</ EditContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditContact >
  <Status> Failure </Status>
  <Details>
    <Message>You have reached the user limit. Please contact PWF support to upgrade. </Message>
  </Details>
</ EditContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditContact >
  <Status> Failure </Status>
  <Details>
    <Message> Your Username/Password combination is assigned to another user. Please try again
    with an alternative combination. </Message>
  </Details>
</ EditContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditContact >
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ EditContact >
```

## AddCompany- Adds a company to the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addcompany](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addcompany)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< NewCompany>
  < UserID>5</ UserID >
  < ContactTypeID>3</ ContactTypeID>
  < ContactCatID>1</ ContactCatID>
  < CompanyName>ProActive Software Ltd</ CompanyName>
  < CompanyCode>PWF</ CompanyCode>
  < WorkPhone>+643 347 3221</ WorkPhone>
  < Fax> +643 347 3221</ Fax>
  < EmailAddress>sales@proworkflow.com</ EmailAddress>
  < Address1>328 Two Chain Road</ Address1>
  < Address2>Rolleston</ Address2>
  < Address3> </ Address3>
  < City>Christchurch</ City>
  < State></ State>
  < ZipCode>7675</ ZipCode>
  < Country> New Zealand</ Country>
  < WebsiteURL>www.proworkflow.com</ WebsiteURL>
  <Notes>Testing</ Notes>
</ NewCompany >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – None
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AddCompany >
  <Status>Success</Status>
  <Details>
    <Message> Company added</Message>
    <CompanyID>34</ CompanyID >
  </Details>
</ AddCompany >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddCompany >
  <Status> Failure </Status>
  <Details>
    <Message> Error: You don't have permission to perform this action..</Message>
  </Details>
</ AddCompany >
```

## EditCompany- Edits a company in the database

1. Web service call - e.g.

[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=editcompany](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=editcompany)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< EditCompany>
  < UserID>5</ UserID >
  < ContactTypeID>3</ ContactTypeID>
  < ContactCatID>1</ ContactCatID>
  < CompanyID>34</ CompanyID >
  < CompanyName>ProActive Software Ltd</ CompanyName>
  < CompanyCode>PWF</ CompanyCode>
  < WorkPhone>+643 347 3221</ WorkPhone>
  < Fax> +643 347 3221</ Fax>
  < EmailAddress>sales@proworkflow.com</ EmailAddress>
  < Address1>328 Two Chain Road</ Address1>
  < Address2>Rolleston</ Address2>
  < Address3> </ Address3>
  < City>Christchurch</ City>
  < State></ State>
  < ZipCode>7675</ ZipCode>
  < Country> New Zealand</ Country>
  < WebsiteURL>www.proworkflow.com</ WebsiteURL>
  <Notes>Testing</ Notes>
</ EditCompany >
```

Required variables are highlighted in red. Fields that are not necessary can be removed from the Request Data Packet except the mandatory fields.

4. URL Parameters – None

5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.

6. Status messages include: Success, Failure or Error

7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< EditCompany >
  <Status>Success</Status>
  <Details>
    <Message> Company edited</Message>
    <CompanyID>34</ CompanyID >
  </Details>
</ EditCompany >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< EditCompany >
  <Status> Failure </Status>
  <Details>
    <Message> Error: You don't have permission to perform this action..</Message>
  </Details>
</ EditCompany >
```

**DeleteContact** - Deletes a contact and insert in the audit log.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=deletecontact&ContactID=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=deletecontact&ContactID=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – ContactID and UserID are mandatory.
4. Note - User Permissions are also checked for while deleting the contacts.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteContact>
  <Status>Success</Status>
  <Details>
    <Message>Contact deleted</Message>
  </Details>
</ DeleteContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteContact >
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ DeleteContact >
```

**DeleteCompany** - Deletes a Company and insert in the audit log.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=deletecompany&CompanyID=xxx&UserID=xxx&DeleteAction=DeleteCompany](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=deletecompany&CompanyID=xxx&UserID=xxx&DeleteAction=DeleteCompany)
2. Method - GET
3. URL Parameters – CompanyID and UserID are mandatory. DeleteAction is not required but if specified the API will act on that basis. DeleteAction can be either set to “DeleteCompany” or “DeleteAll”
4. Note - User Permissions are also checked for while deleting the company.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteContact >
    <Status>Success</Status>
    <Details>
        <Message> Company Record deleted successfully. </Message>
    </Details>
</ DeleteContact >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteContact >
    <Status> Failure </Status>
    <Details>
        <Message>Error: You don't have permission to perform this action.. </Message>
    </Details>
</ DeleteContact >
```

## AddContactCategory - Adds a contact category to the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addcontactcategory](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addcontactcategory)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< NewContactCategory>
  < UserID>5</ UserID >
  < CategoryName>Current Projects</ CategoryName>
  < DisplayOrder>3</ DisplayOrder>
</ NewContactCategory>
```

4. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.

5. Status messages include: Success, Failure or Error

6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContactCategory>
  < Status>Success</Status>
  < Details>
    < Message> Contact category added</Message>
    < ContactCatID>7</ContactCatID></Details>
</ AddContactCategory>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddContactCategory>
  < Status> Failure </Status>
  < Details>
    < Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ AddContactCategory>
```

GetContactDetailsFromIdList – Retrieves all the contact details for the contacts assigned to a job or tasks.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=contactlist&ContactIDList=x5x](https://proworkflow2.net/pwfaccountname/api/contacts.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=contactlist&ContactIDList=x5x)
2. Method - GET
3. URL Parameters – ContactIDList is mandatory

4. Note : Just pass the whole list of contacts with the delimiters and a XML packet will be returned with details of each contacts similar to the GetSpecificContactDetails method
5. Return Data – A XML data packet of ContactsList detail is constructed by the web service and sent to the user.
6. Status messages include: OK or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< Contacts>
  <Status>OK</Status>
  <Contact_Count>1</Contact_Count>
  <ContactsList>
    <Contact>
      <ContactId>18</ContactId>
      <ContactCategoryId>1</ContactCategoryId>
      <ContactTypeID>3</ContactTypeID>
      <CompanyId>5</CompanyId>
      <CompanyName><![CDATA[Big Company]]></CompanyName>
      <ContactJobTitle>CEO</ContactJobTitle>
      <FullName>Aubrey Apple</FullName>
      <AddressLine1>328 Two Chain Road</AddressLine1>
      <AddressLine2>Rolleston</AddressLine2>
      <AddressLine3> </AddressLine3>
      <WorkPhone>+643 347 3221</WorkPhone>
      <MobilePhone>+6421 347 3223</MobilePhone>
      <HomePhone>+6421 347 3223</HomePhone>
      <Fax>+6421 347 3225</Fax>
      <AllowLogin>1</AllowLogin>
      <LoginName>demo</LoginName>
      <Password>demo</Password>
      <PrimaryContact>0</PrimaryContact>
      <PhoneExtension>ext</PhoneExtension>
      <EmailAddress/>
      <City>Christchurch</City>
      <State/>
      <ZipCode>7675</ZipCode>
      <Country> New Zealand</Country>
    </Contact>
  <ContactsList>
</ Contacts>
```

## Time Manager API

---

**AddTimeRecord**- Inserts a Time record in the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=addtimerecord](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=addtimerecord)

2. Method - POST

3. XML Request Data – e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
< NewTimeRecord>
  < UserID>5</ UserID >
  < TaskID>6</ TaskID>
  < TrackerDate>1-May-2009 8:43:00 AM</ TrackerDate>
  < Notes>6</ Notes>
  < TrackerTimeFinished>1-May-2009 8:43:00 AM</ TrackerTimeFinished>
  < TrackerTimeStarted>1-May-2009 8:20:00 AM</ TrackerTimeStarted>
</ NewTimeRecord>
```

4. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
5. Note: TrackerDate, TrackerTimeFinished and TrackerTimeStarted should be in sql dateformat type
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTimeRecord>
  <Status>Success</Status>
  <Details>
    <Message> Time Record Added</Message>
    <TrackerID>7</TrackerID></Details>
</ AddTimeRecord>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTimeRecord>
  <Status> Failure </Status>
  <Details>
    <Message>You have entered a Finish time which is BEFORE the Start time.</Message>
  </Details>
</ AddTimeRecord>
```

OR

```

<?xml version="1.0" encoding="UTF-8"?>
< AddTimeRecord>
    <Status> Failure </Status>
    <Details>
        <Message>Error: You don't have permission to perform this action.. </Message>
    </Details>
</ AddTimeRecord>

```

**EditTimeRecord**- Inserts a Time record in the database

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=edittimerecord](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=edittimerecord)

2. Method - POST

3. XML Request Data – e.g.

```

<?xml version="1.0" encoding="UTF-8"?>
< EditTimeRecord>
    < UserID>5</ UserID >
    < TrackerID>7</ TrackerID>
    < TrackerDate>1-May-2009 8:43:00 AM</ TrackerDate>
    < Notes>Finished last part of this task</ Notes>
    < TrackerTimeFinished>1-May-2009 8:43:00 AM</ TrackerTimeFinished>
    < TrackerTimeStarted>1-May-2009 8:20:00 AM</ TrackerTimeStarted>
</ EditTimeRecord>

```

4. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
5. Note: TrackerDate, TrackerTimeFinished and TrackerTimeStarted should be in sql dateformat type
6. Status messages include: Success, Failure or Error
7. Example of return data:

```

<?xml version="1.0" encoding="UTF-8"?>
< EditTimeRecord>
    <Status>Success</Status>
    <Details>
        <Message> Time Record Edit</Message>
        <TrackerID>7</TrackerID></Details>
</ EditTimeRecord>

```

OR

```

<?xml version="1.0" encoding="UTF-8"?>
< AddTimeRecord>

```

```
<Status> Failure </Status>
<Details>
  <Message>You have entered a Finish time which is BEFORE the Start time.</Message>
</Details>
</ AddTimeRecord>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< AddTimeRecord>
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ AddTimeRecord>
```

**DeleteTimeRecord**- Deletes a Time record.

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=deletetimerecord&TrackerID=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=deletetimerecord&TrackerID=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – TrackerID and UserID are mandatory
4. Note - User Permissions are also checked for while deleting the time record.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteTimeRecord >
  <Status>Success</Status>
  <Details>
    <Message> Time Record deleted successfully. </Message>
  </Details>
</ DeleteTimeRecord >
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< DeleteTimeRecord >
  <Status> Failure </Status>
  <Details>
    <Message>Error: You don't have permission to perform this action.. </Message>
  </Details>
</ DeleteTimeRecord >
```

**GetTimeRecords**- Retrieves all the time records for a task

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=timerecords&TaskID=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=timerecords&TaskID=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – TaskID is mandatory. If userid not specified then default is to show information for all.
4. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
5. Status messages include: OK or Error
6. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< TaskTimeRecords >
  <Status>OK</Status>
  < TimeRecord_Count>1</ TimeRecord_Count >
  <TimeRecords>
    <TimeRecord>
      <id>21847</id>
      < notes>test</ notes >
      < contactid >5</ contactid >
      <trackerdate>20-Jul-2009</trackerdate>
      <activetimer>0</activetimer>
      <timestarted>15:27</timestarted>
      <timefinished>15:27</timefinished>
      <hours>0</hours>
      <minutes>0</minutes>
      <deleted>0</deleted>
      <paused></paused>
    </ TimeRecord >
  </ TimeRecords>
</ TaskTimeRecords >
```

**StartTimer**- Starts a Timer for a Task

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=starttimer&TaskID=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=starttimer&TaskID=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – TaskID and Userid are mandatory.
4. Note - User Permissions are also checked for while deleting the company.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error

7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< StartTimer>
  <Status> Success</Status>
  <Details>
    <Message> Timer Started </Message>
    <TrackerID>3679</TrackerID>
  </Details>
</ StartTimer>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< StartTimer>
  <Status> Success</Status>
  <Details>
    <Message>Error: You don't have permission to perform this action..</Message>
  </Details>
</ StartTimer>
```

### StopTimer- Stops the Timer

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=starttimer&TrackerID=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/time.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=starttimer&TrackerID=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – TrackerID and Userid are mandatory.
4. Note - User Permissions are also checked for while deleting the company.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: Success, Failure or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
< StopTimer>
  <Status> Success</Status>
  <Details>
    <Message> Timer Stopped </Message>
    <TrackerID>3679</TrackerID>
  </Details>
</ StopTimer>
```

OR

```
<?xml version="1.0" encoding="UTF-8"?>
< StopTimer>
  <Status> Success</Status>
  <Details>
    <Message>Error: You don't have permission to perform this action..</Message>
  </Details>
</ StopTimer>
```

## Upcoming Items API

---

**GetUpcomingItems**- Retrieves the Upcoming Items for a given timeframe

1. Web service call - e.g.  
[https://proworkflow2.net/pwfaccountname/api/upcoming.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api\\_call=upcomingitems&startdate=xxx&enddate=xxx&UserID=xxx](https://proworkflow2.net/pwfaccountname/api/upcoming.cfm?customerkey=84DC6CF8-BF78-D79D-B25B-0C0D20229F17&api_call=upcomingitems&startdate=xxx&enddate=xxx&UserID=xxx)
2. Method - GET
3. URL Parameters – Userid, startdate and enddate are not mandatory.
4. Note – If no userid is specified then show information for all is used as a default. If no startdate is specified then the current date will be used. If no enddate is specified then it is set to 2 weeks from the startdate.
5. Return Data – A XML data packet is constructed by the web service and sent to the user. It contains the success or failure message.
6. Status messages include: OK or Error
7. Example of return data:

```
<?xml version="1.0" encoding="UTF-8"?>
<UpcomingItems>
  <Status>OK</Status>
  <Item_Count>3</Item_Count>
  <ItemList>
    <Item>
      <id>939</id>
      <title>Test Job</title>
      <number>12345</number>
      <startdate>02-Feb-2009</startdate>
      <duedate>13-Mar-2009</duedate>
      <status>Job Overdue</status>
      <type>Job</type>
      <Job>
        <id>939</id>
        <title>Test Job</title>
        <client>Big Company</client>
        <number>12345</number>
        <Startdate>02-Feb-2009</Startdate>
        <Duedate>13-Mar-2009</Duedate>
        <Completeddate>27-Jun-2009</Completeddate>
        <Invoiced>N</Invoiced>
        <Paid>N</Paid>
        <AccountedFor>N</AccountedFor>
        <AllocatedTime>426</AllocatedTime>
        <TotalTimeSpent></TotalTimeSpent>
        <JobComplete>0</JobComplete>
        <JobDescription>Test test</JobDescription>
        <Notes> </Notes>
        <JobStatus>14</JobStatus>
      </Job>
    </Item>
  </ItemList>
</UpcomingItems>
```

```

    <JobAssignedTo>x5</JobAssignedTo>
    <JobAllocatedhours>7</JobAllocatedhours>
    <JobAllocatedminutes>6</JobAllocatedminutes>
    <Quotedprice> 5234.625</ Quotedprice>
    <ClientLogin>1</ClientLogin>
    <InvoicedPrice>798.75</InvoicedPrice>
    <JobCreatedBy>5</JobCreatedBy>
    <JobCategory>Production</JobCategory>
    <JobManager>5</JobManager> (Job Creator)
    <JobManagerDetails>
      <ContactId>5</ContactId>
      <ContactCategoryId>1</ContactCategoryId>
      <ContactTypeId>3</ContactTypeId>
      <CompanyId>Big Company</CompanyId>
      <FullName>Aubrey Apple</FullName>
      <AddressLine1>328 Two Chain Road</AddressLine1>
      <AddressLine2>Rolleston</AddressLine2>
      <AddressLine3> </AddressLine3>
      <WorkPhone>+643 347 3221</WorkPhone>
      <MobilePhone>+6421 347 3223</MobilePhone>
      <EmailAddress/>
      <City>Christchurch</City>
      <State/>
      <ZipCode>7675</ZipCode>
      <Country> New Zealand</Country>
    </JobManagerDetails>
  </Job>
</Item>
<Item>
  <id>1567</id>
  <title>Test Task</title>
  <number></number>
  <startdate>02-Feb-2009</startdate>
  <duedate>13-Mar-2009</duedate>
  <status>Task Overdue</status>
  <jobid>939</jobid>
  <type>Task</type>
  < Task>
    <taskid>1567</taskid>
    <jobid>939</jobid>
    <title>Test Task</title>
    <taskstatus>A </taskstatus>
    <datetostart>02-Feb-2009</datetostart>
    <datedue>13-Mar-2009</datedue>
    <taskcreatedby>5</taskcreatedby>
    <taskassignedto> x53x1x </taskassignedto>
    < Taskpriority>3</ Taskpriority>
    < allocatedtime>0/ allocatedtime>
    < deleted>0</ deleted>
  </ Task>
</Item>

```

```

        <compdate> </ compdate>
        < taskorder >9.0</ taskorder >
        <taskbilling>1</taskbilling>
        <taskdescription></taskdescription>
        <allocatedtime_hours>3</allocatedtime_hours>
        <allocatedtime_minutes>0</allocatedtime_minutes>
        <totaltimespent>3:12</totaltimespent>
        <taskstyle_bh>1</taskstyle_bh>
        <taskstyle_b>0</taskstyle_b>
        <tracking>1</tracking >
        <trackerid>157</ trackerid >{this data is only provided if the tracking field is
set to 1}
    </Task>

</Item>
<Item>
    <id>56</id>
    <title>Test Event</title>
    <number></number>
    <startdate>02-Feb-2009</startdate>
    <duedate></duedate>
    <status>Start Event</status>
    <type>Event</type>
</Item>
</ItemList>
</UpcomingItems>

```

## Appendix

---

- Date format: dd-mmm-yyyy
- Time format: hh:mm
- Contact ID Type:
  - 1 = Client
  - 2 = Contractor
  - 3 = Staff
  - 4 = Other
- Variables:
  - 1 = Yes, 0 = No
  - Y = Yes, N = No
  - Task Status: A = Archived, C = Current
  - Job Complete: A = Active, C = Complete
- JobStatus is the ID of the Job Category
- JobClient ID, if the client is a company then append COM in front of the id or if a person then PER in front of the id
- Deleted Contacts are deleted but retain the company, historic data is shown in reports.
  - Delete Company:
    - DeleteCompany the company is deleted but retains any linked people.
    - DeleteAll deletes the company and any linked people.
- Upcoming Items statuses:
  - Job Overdue
  - Job Due
  - Start Job
  - Task Overdue
  - Task Due
  - Start Task
  - Start Event
- For (Get) method status = OK or Error
- For (Post) method status = Success, Failure or Error