appiyo

How to use Appiyo

© 2014 **All rights reserved. All trademarks acknowledged**

This document is published by Appiyo Technologies Pte Ltd. without any warranty.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without the written permission of Appiyo Technologies Pte Ltd.

Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to software programs and/or equipment, may be made by at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Any hard copies of this document are to be regarded as temporary reference copies only.

Table of Contents

© Copyright 2014 Appiyo. All rights reserved.

© Copyright 2014 Appiyo. All rights reserved.

© Copyright 2014 Appiyo. All rights reserved.

About

This document is designed to help you understand Appiyo features with ease. The following sections will guide you through the powerful features and make it an interesting experience.

Join Appiyo

Any individual can independently become an Appiyo member. It is free! To become an Appiyo member, follow the steps below:

1. Go to www.appiyo.com and click **Join Appiyo.**
2. Fill-in the form with your name, email id and password.
3. Click **Join Appiyo**.
4. In order to activate your Appiyo account, please click the activation link received in your email id given.

A group of individuals can decide to form a collaborative team. Every individual must be an independent Appiyo member first. A representative of the group can then create a Group in Appiyo and invite others to join the group.

A company interested in using Appiyo must first instruct its employees to become independent members using their company email ids. Following this, different interest groups can be formed. During the evaluation stage, it would be advisable for a primary group of individuals in the company to first become members and experience Appiyo. They may subsequently choose to co-opt other co-employees also, by inviting them to become Appiyo members.

Sign in

To sign into your Appiyo account:

1. Go to top of www.appiyo.com and click **Sign In**.
2. Enter your Appiyo account credentials (**email id** and **password**).
3. Click **Sign in**.

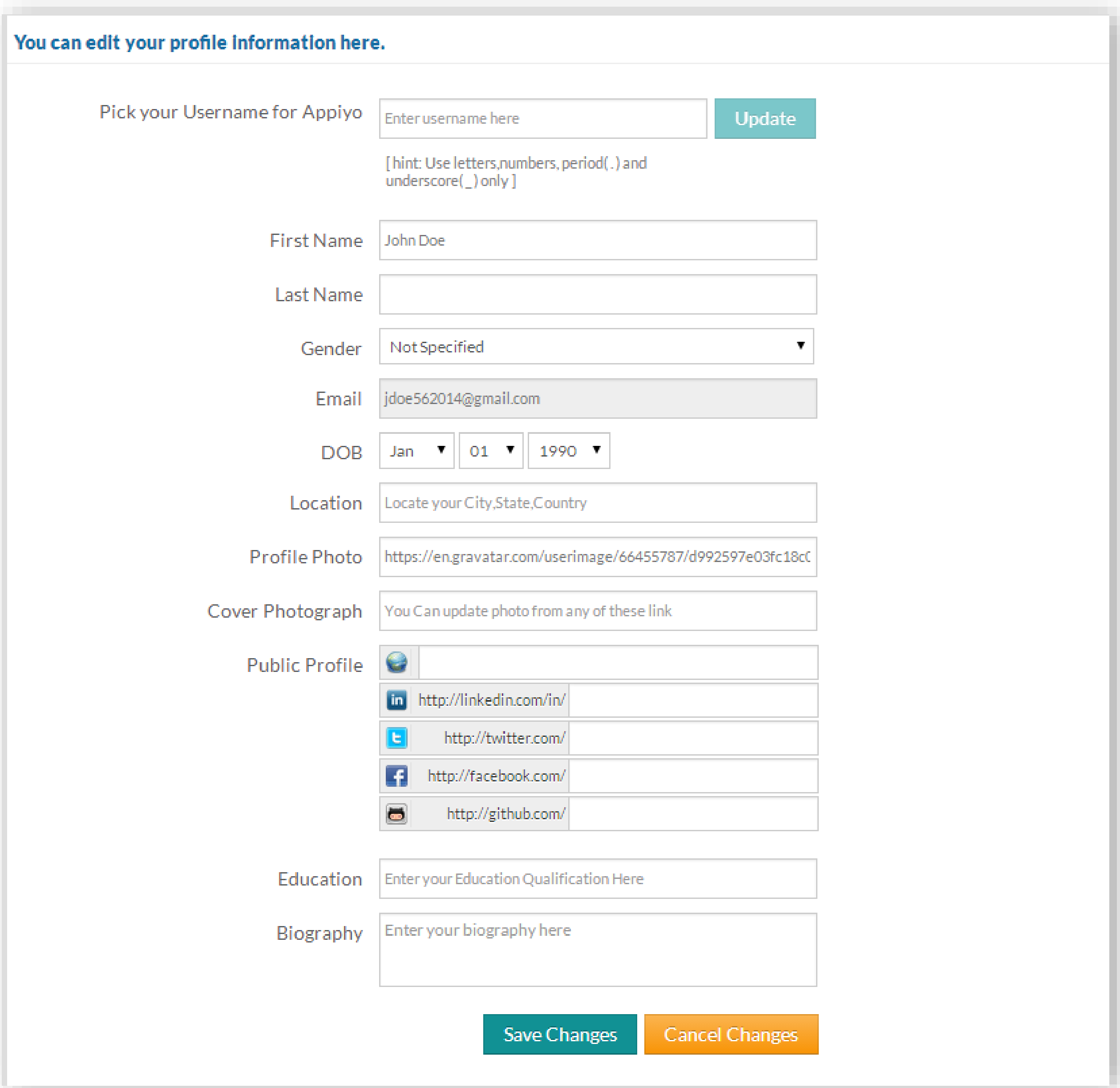
Setup Profile

Setup User Profile

Adjacent to your profile photo, is a banner space provided. Here you can post a thematic picture, just to lend a personality to your profile page and keep it interesting. Additionally, you get to know which groups and processes you have created and which are those where you have been co-opted as a member.

Navigation: Your Name (near the menu at top right) → My Profile → Edit Profile (under cover photo)

© Copyright 2014 Appiyo. All rights reserved.



To update your user profile, click **Edit Profile** under cover photo and make necessary changes.

To add a profile photo, first create an account in www.gravatar.com and include the link in the **Profile Photo** field.

To change the background picture of the profile, add photo in the **Cover Photograph** field.

After completing all necessary modifications, click **Save Changes.**

© Copyright 2014 Appiyo. All rights reserved.

Setup Zone Profile

Zones are a powerful feature in Appiyo. They are clearly demarcated virtual organizations on cloud. They provide the first level of privacy and security. A zone is a virtual boundary under the control of the zone administrator who controls and secures access to the zone and resources within it.

Zones are of two types. One is a ‘community zone’ and the other is a ‘commercial zone’.

Community zones are lifetime free. They may also receive ads.

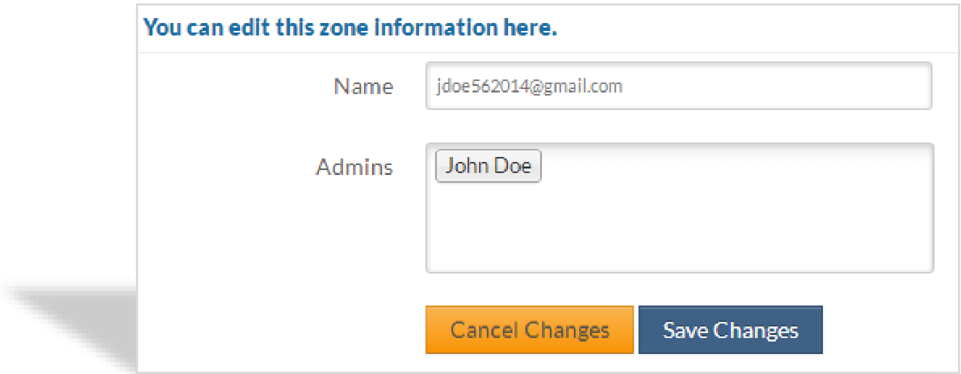
On sign-up, every member gets her/his own exclusive community zone. Every member can have a personal and unique experience in her/his exclusive community zone.

When a member wants to go commercial with her/his solutions, they have to opt for commercial zones. These zones will be ad-free.

> Example A, a BPM training company can opt for a commercial zone, create different groups (Batches) inside the zone and invite trainees into those batches for the duration of the program and close the batch upon program completion.
>
> Example B, a company is interested in a solution created by an Appiyo member. The company should create a commercial zone and the solution will be deployed into that zone.

By default, the primary member of a zone will have administrator privilege which will allow the member to operate her/his zone the way she/he wants.

Navigation: Admin → Edit Zone (under cover photo)



To **update your zone profile**, click **Edit Zone** under cover photo.

Enter your **Name** and enter your zone member names to add them as your zone **Admins**.

Click **Save Changes**.

Note: New Appiyo users can click **Let’s get started** button on **Home** page to update their zone and user profile.

© Copyright 2014 Appiyo. All rights reserved.

Zone Level Groups

Zone is a secure boundary provided for all Appiyo members within which the zone members can form a closed group (like a firm or company).

Group creation	Any zone member can form the group. Group can be public, private or restricted.
Member participation	Only zone members can participate. There are different ways by which you can participate in a group:- ▪ by subscribing to a group of your interest (**Public**). ▪ by sending a request to join a group and waiting for approval from that group owner (**Private**). ▪ by receiving an invite from a group owner (**Restricted**).
Group Invite	**Zone Members** Group owner can send invitation to zone members and they can accept and subscribe. **Appiyo Members** ▪ First the zone administrator will add the invitee as zone member (Admin/Users/Invite). ▪ Group owner can send invitation to the zone member and she/he can accept and subscribe. **Non Appiyo Members** ▪ Appiyo Membership activation link will be sent to the invitee's email id given by the group owner. ▪ Invitee can accept and become an Appiyo member. ▪ The zone administrator will add the invitee as zone member (Admin/Users/Invite). ▪ Now the group owner can invite to the group.

Appiyo Level Groups

Appiyo Groups is a collection of Appiyo members within which the members (zone and Appiyo) can collaborate with others outside their zone.

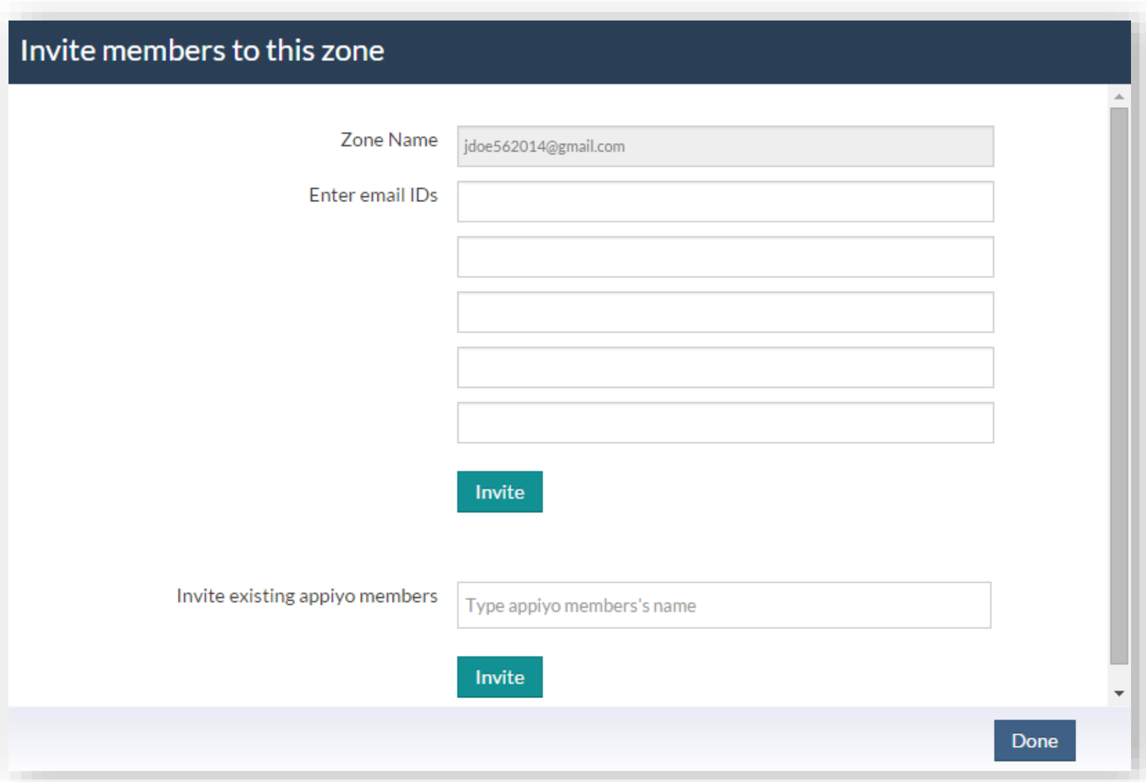
Group creation	Only zone admins can form Appiyo level groups. Group can be public, private or restricted.
Member participation	All Appiyo members can participate and it doesn't require zone membership. There are different ways by which you can participate in a group:- ▪ by subscribing to a group of your interest (**Public**). ▪ by sending a request to join a group and waiting for approval from that group owner (**Private**). ▪ by receiving an invite from a group owner (**Restricted**).
Group invite (zone membership is not required)	**Appiyo Members** ▪ Group owner can send invitation to the members and they can accept and subscribe. **Non Appiyo Members** ▪ Appiyo Membership activation link will be sent to the invitee's email id given by the group owner. ▪ Invitee can accept and become an Appiyo member and group member.

© Copyright 2014 Appiyo. All rights reserved.

My Zone users

Here you can invite existing users or can create new users to join your zone. Only zone owners can invite others to subscribe to their zone.

Navigation: Admin tab → Users tab



To **add a new user or invite existing users** to your zone, click **Invite**.

Enter email IDs to add new user. Activation link will be sent to the specified email id to activate the account of the new user.

If the user is already an Appiyo member, **Type appiyo member's name** and click **Invite**. Once the member accepts the invite (sent as notification), he will be added to your zone.

Note: To switch between subscribed zones, click on **My Zone** at top right.

To set a default zone, click on your name at top right and click **Set Default Zone**.

© Copyright 2014 Appiyo. All rights reserved.

Groups

- You get a helicopter view of various Appiyo groups created by members. Some 'tag' information is also provided. Appiyo **Groups** provides a vibrant private ecosystem for effective and secure exchange of thoughts and ideas, some of which can progress towards implementation, business and monetization.
- You can '**Create a Group**' for different purposes.
  - If you create a **public** group, any Appiyo member can join the group. If you create a **private** group, any Appiyo member wanting to join will need your permission.
  - If you create a **restricted** group, it will not be visible to others. Only members you explicitly invite can join.
  - Only a zone administrator can form groups which have the full Appiyo ecosystem members as scope for membership. For any member in a zone who is not the zone's administrator, the scope for member participation is restricted to members of only that zone.
  - Typically we can wonder "what could possibly be interesting about Groups?" In addition to providing security and privacy, Groups provide a way to collaborate at different levels depending on peculiar sensitivities involved.
  - For example, public groups can be set up for an organization wide involvement. Private groups can narrow down the membership to discuss specific issues or feedbacks. This can be priceless because it can bring valuable perspectives. Restricted groups can be restricted to issues concerning strategy and any confidential matters. Some examples are C-level initiatives, C-level succession planning, senior level appraisals and the like.
- You can click on a group to get to group details (also refer to Groups).

Social Group and Work Group

There are two types of groups: **Social Groups** and **Work Groups**.

In social groups you can invite other members in your zone or across zones or outside Appiyo for discussions. Social groups can be **zone social** or **Appiyo social groups**.

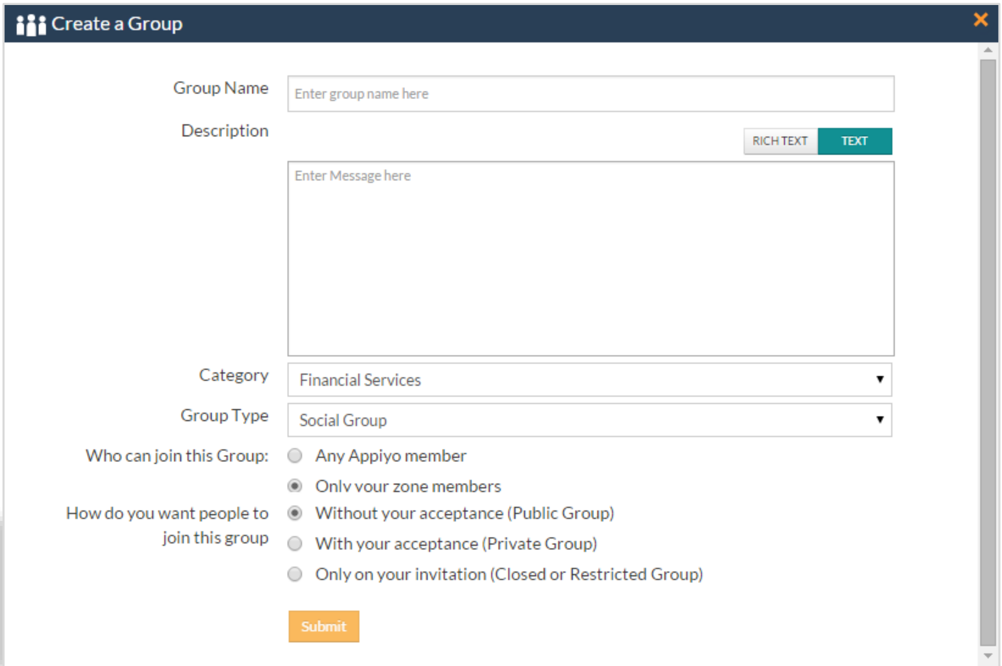
Work groups are created by the zone admins to assign tasks (refer to User Task) to the zone users. Work groups are restricted to only **zone** level.

Note: Social groups that are created in other subscribed zones will be **zone social groups** by default.

© Copyright 2014 Appiyo. All rights reserved.

Create Group

Navigation: Groups tab → Create a Group



To create a group, click **Create a Group** under **Groups**.

Enter a name for the group a description about the group and set the group type as **Social Group** or **Workgroup.**

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

> **RICH TEXT** – Includes formatting and hyperlink attachment options.
>
> **TEXT** – Ignores formatting and hyperlink attachment options.

For **Social Group**, select a category from the drop-down list and set the members type (**zone** or **Appiyo**) who can join your group.

'**Only your zone members**' social group can be viewed only by your zone users and '**Any Appiyo member**' social group can be viewed by all the Appiyo members across zones.

Set (**Public**, **Private or Closed/Restricted**) to restrict the joining process.

'**Without your acceptance (Public Group)**' allow all users in your zone to view the social group and they themselves can join the group.

'**With your acceptance (Private Group)**' allows only subscribed persons to view the social group. But it will be included in the group list and zone users need your permission to join the group.

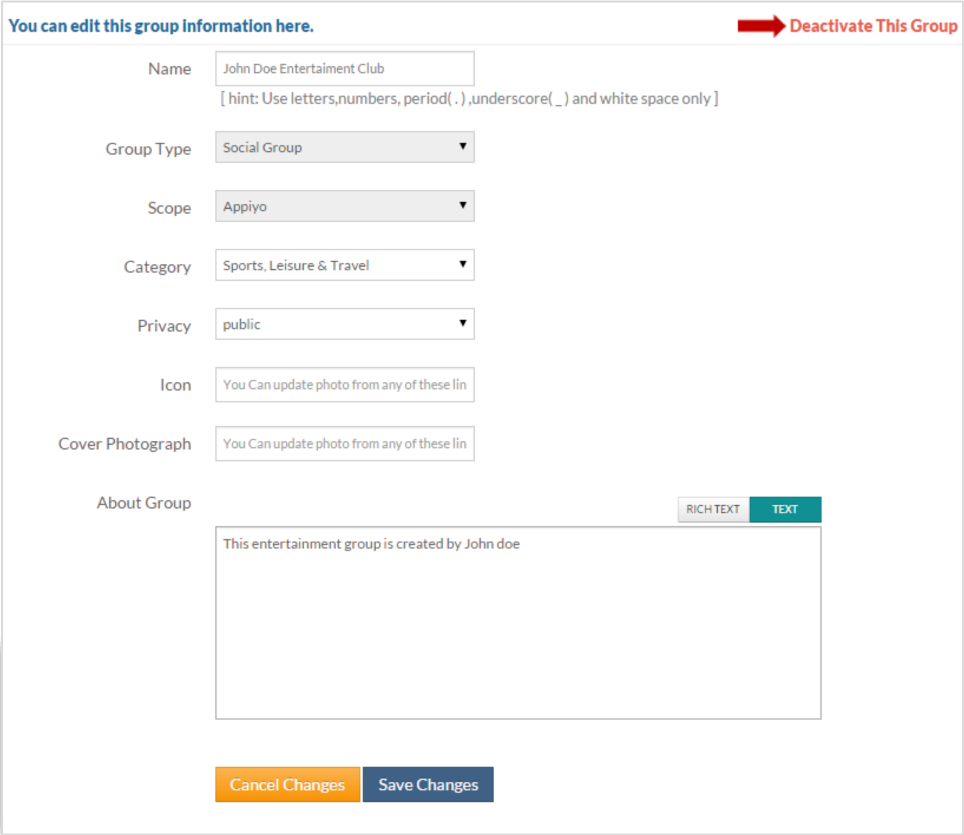
'**Only on your invitation (Closed or Restricted Group)**' allows only the owner and subscribed persons to view the social group and it will not be displayed in the list of group.

Click **Submit**; you can view the newly created group listed under Groups.

© Copyright 2014 Appiyo. All rights reserved.

Edit Group allows you to modify your group details.

Navigation: Required Group → Description tab → Edit Group (under group cover photo)



To **modify your group details**, click **Edit Group** (under group cover photo) under **Description** tab of required group, make necessary changes and click **Save Changes.**

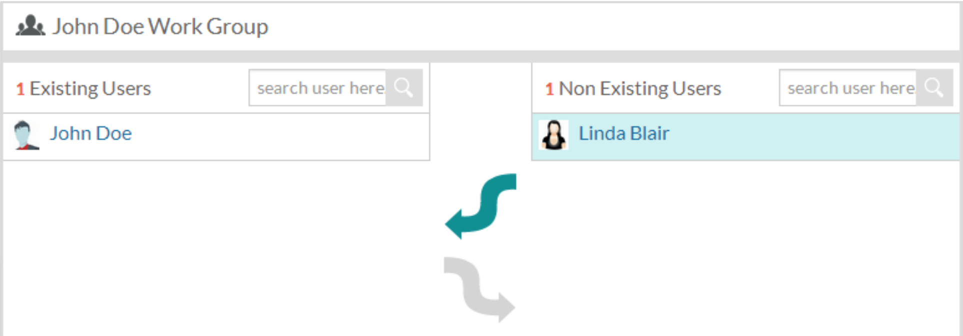
Deactivate This Group allows you to delete your group.

To **delete your group**, click **Deactivate This Group** under **Edit Group** and click **Yes**.

Note: To reactivate the deactivated group, click **Activate this Group** under required group.

Map zone user to work group

Map users allows you to add your zone users to the workgroup for assigning tasks (refer to User Task).

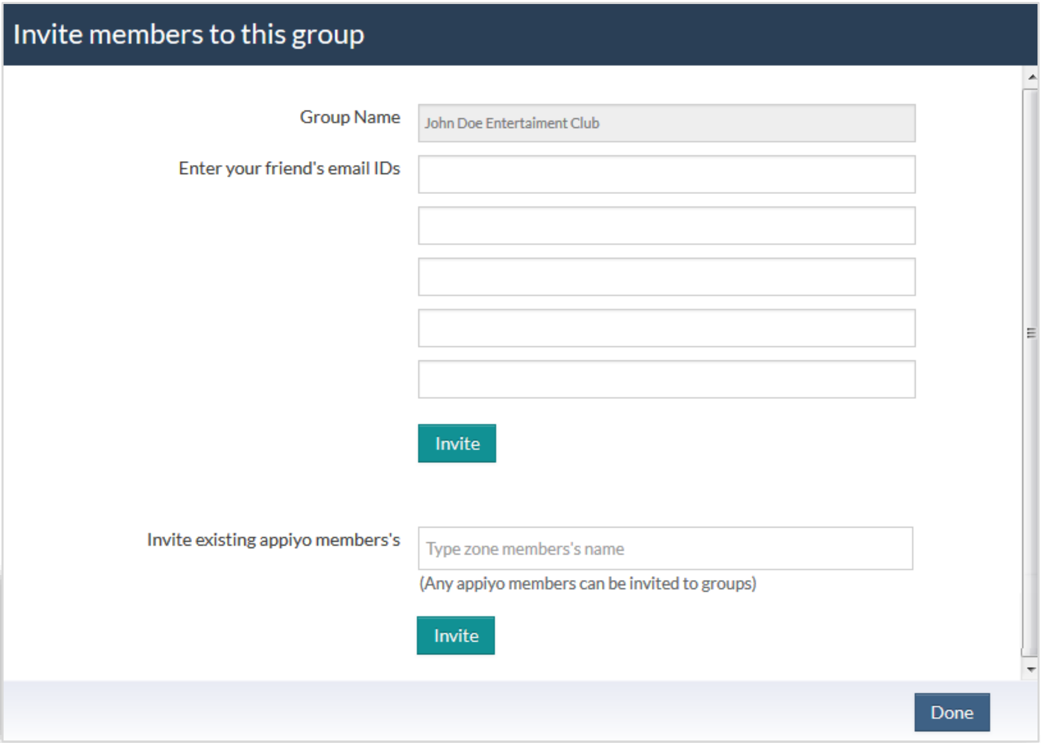


To **map your zone users to** your **workgroup**, click **Map User** under **Description** tab of required group and add **Non Existing Users** on the right to **Existing Users** on the left using the arrows given below.

© Copyright 2014 Appiyo. All rights reserved.

Invite to social group

You can use this feature to invite zone/Appiyo/non-Appiyo users to join your social groups. Only your zone users can be invited to zone social groups whereas non-zone and non-Appiyo users can also be invited to Appiyo social groups.



To **invite non-Appiyo members** to join your social group, click **Invite** button under **Description** tab of required social group, **Enter your friend's email IDs** and click **Invite**.

Once the Appiyo account is activated (activation link will be sent to the given id), she/he will become your group member.

To **Invite existing** a**ppiyo members's** to join your social group, enter the member's name and click **Invite.**

Subscribe to social group

You can use this feature to accept the invitation and subscribe to the social group.

To **subscribe to a social group**, click **Accept** of required social group under **Home** (Notification).

Reject social group invite

Reject a social group invite allows you to reject the invite from a social group.

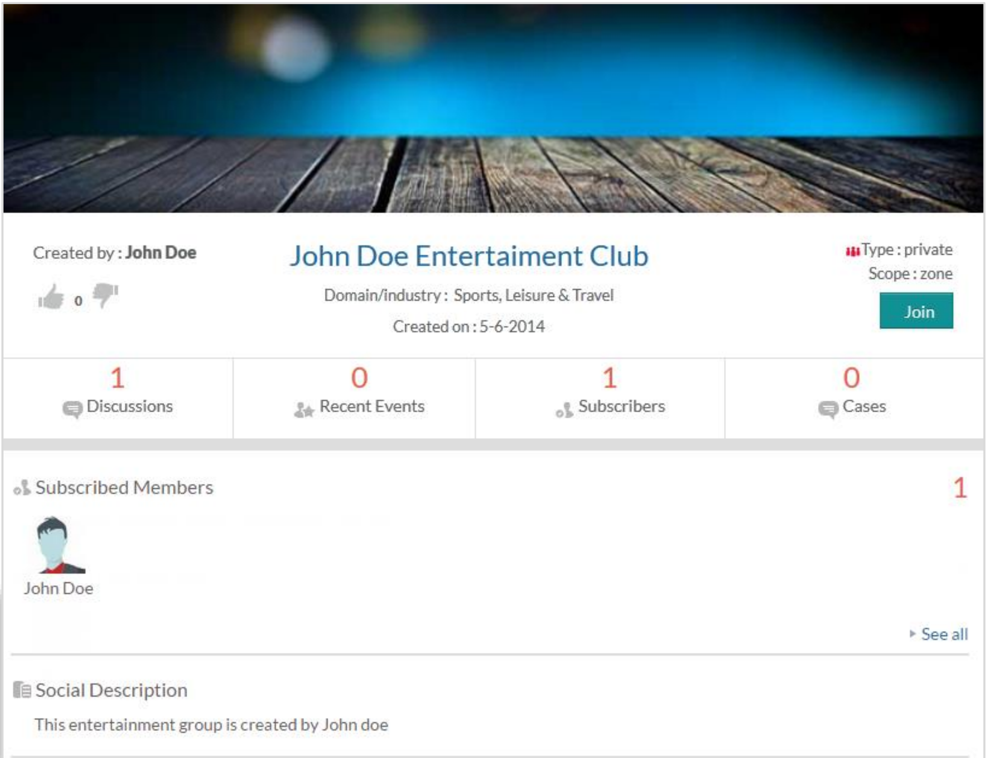
To **reject a social group invite**, click **Reject** of the required social group under **Home** (Notification).

© Copyright 2014 Appiyo. All rights reserved.

Join social group

Join social group (if the group is private) allows you to send group subscription request to its owner.

Navigation: Groups tab → Explore → Required Social Group → Join



To **send subscription request to social group owner**, click **Join** of required social group.

Approve social group subscription request

You can approve the request of other users to join your group.

To **approve social group subscription request**, click **Accept** of required group under **Home** (Notification).

Reject social group subscription request

You can reject the subscription request of other users to join your group.

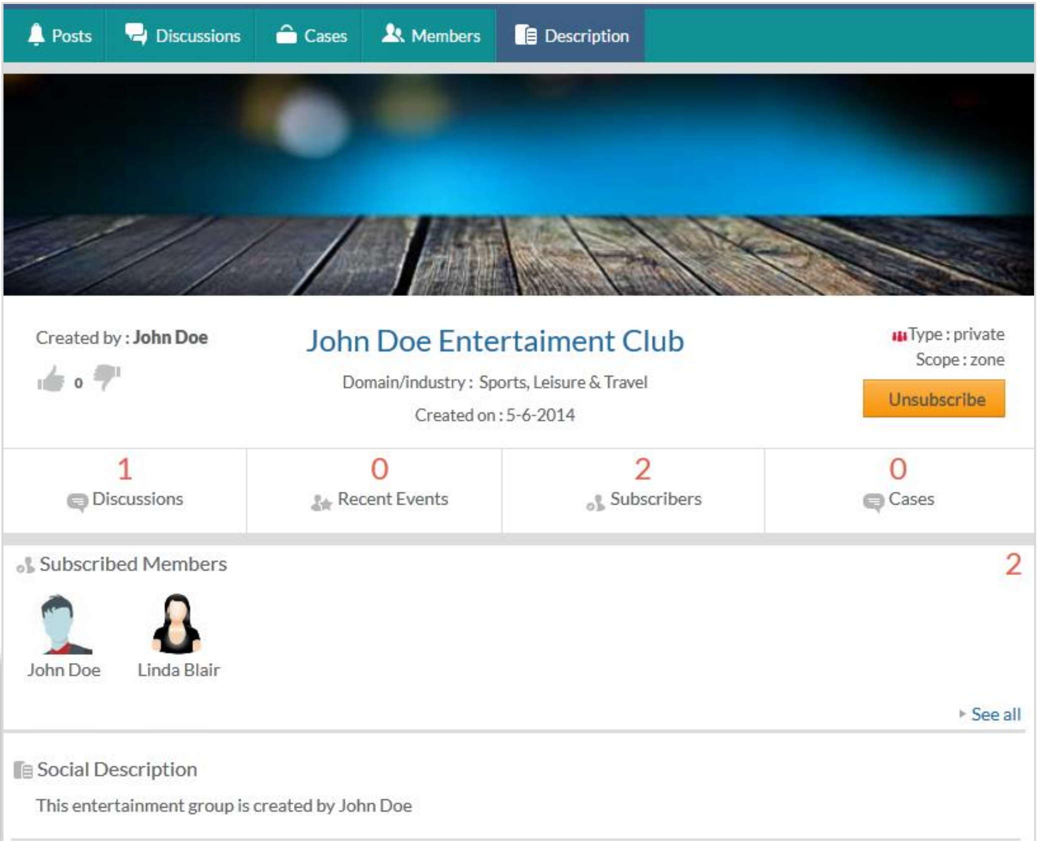
To **reject social group subscription request**, click **Reject** of required group under **Home** (Notification).

© Copyright 2014 Appiyo. All rights reserved.

Unsubscribe social group

Unsubscribe social group allows you to leave the group.

Navigation: Required Social Group → Description tab → Unsubscribe



To **unsubscribe a social group**, click **Unsubscribe** under **Description** tab of required group.

Note: To join or to invite to a zone social group, the user must belong to that zone i.e. the user must be zone user.

Up and Down Vote

Up Vote allows you to add your like to groups.

To **add your like to group** click the **icon** from the required group.

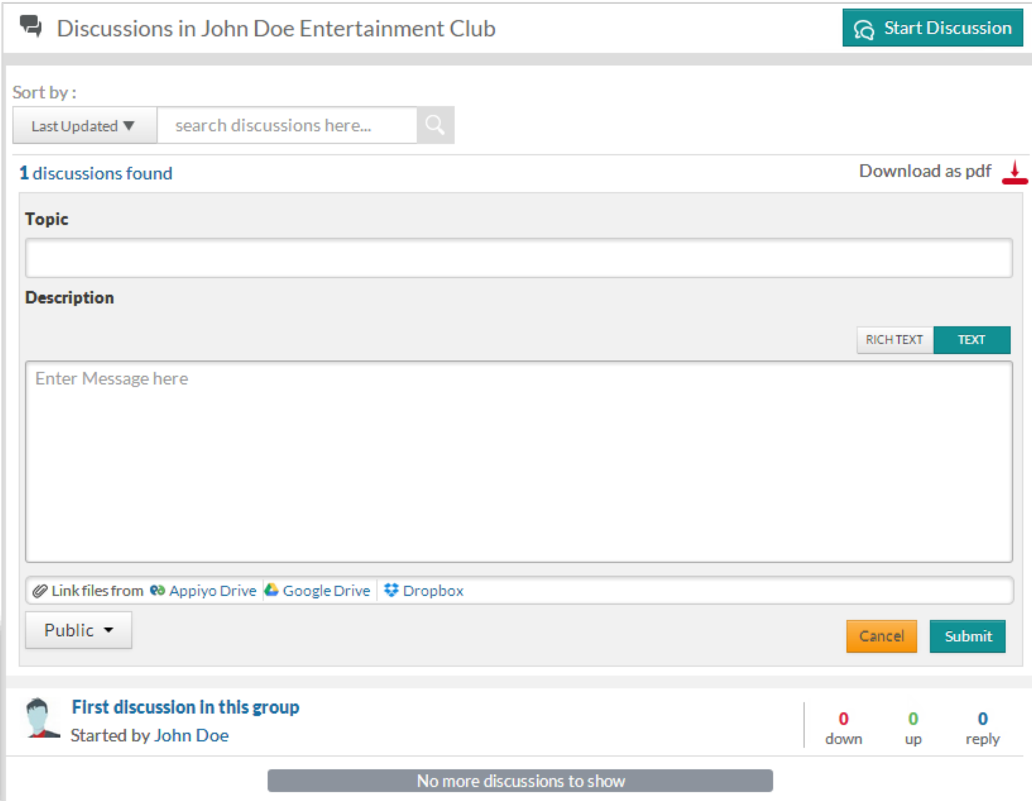
Down Vote allows you to add your dislike to groups.

To **add your dislike to group** click the **icon** from the required group.

© Copyright 2014 Appiyo. All rights reserved.

Discussion

Navigation: Required Group → Discussions tab → Start Discussion



To **start a discussion** for any group, click **Start Discussion** under **Discussion** tab of the required group and enter necessary details under **Topic** and **Description**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

RICH TEXT – Includes formatting and hyperlink attachment options.

TEXT – Ignores formatting and hyperlink attachment options.

Set the type as **Public** or **Private.** You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit**.

To **notify** about a discussion to **all** group **members** (under discussion reply), click **Notify all**.

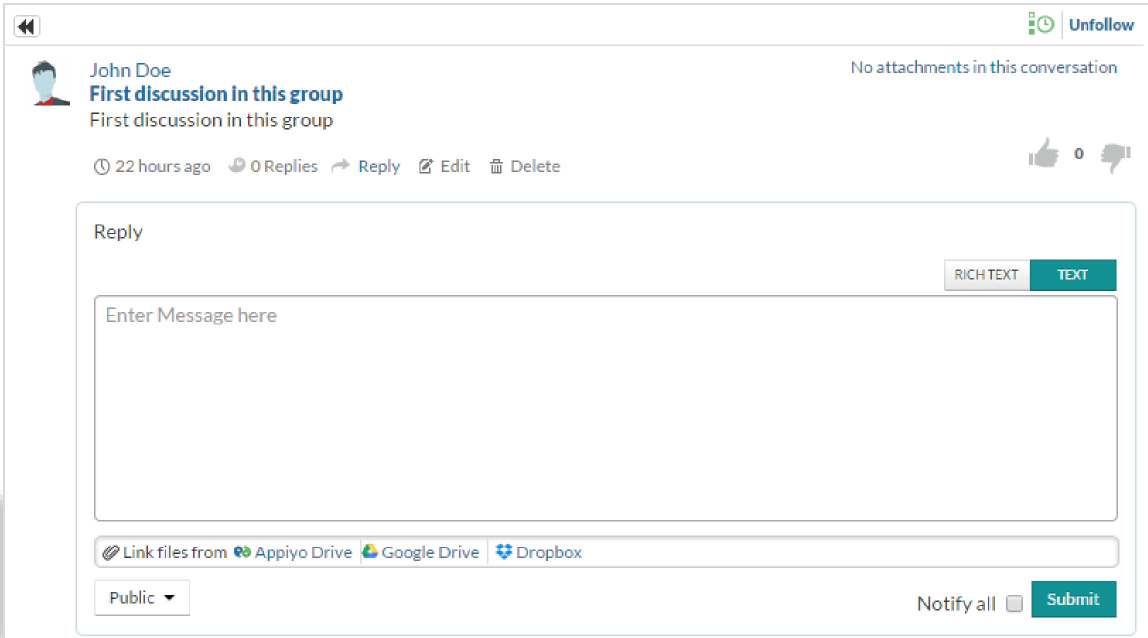
To **notify required members** (under Private discussions), include the members list in the textbox near **Private** button.

To **filter discussions**, search for discussions under **search discussions here**. You can also search for discussions by selecting either of the two options **Last Updated** or **Last Created**.

To **download** all the **discussions as a pdf** file, click **Download as pdf**.

© Copyright 2014 Appiyo. All rights reserved.

To **edit discussion, delete discussion, follow/unfollow discussion** and **up/down vote**, click on the required discussion.



To **modify your discussion**, click **Edit** from the discussion (to be modified), make necessary changes and click **Update**.

To **delete your discussion**, click **Delete** from the discussion (to be deleted) and click **yes**.

To **get notifications about discussions,** click **Follow** of the required discussion.

To **stop getting notifications about discussions,** click **Unfollow** of the required discussion.

To **add your like to discussion** click the **icon** from the required discussion.

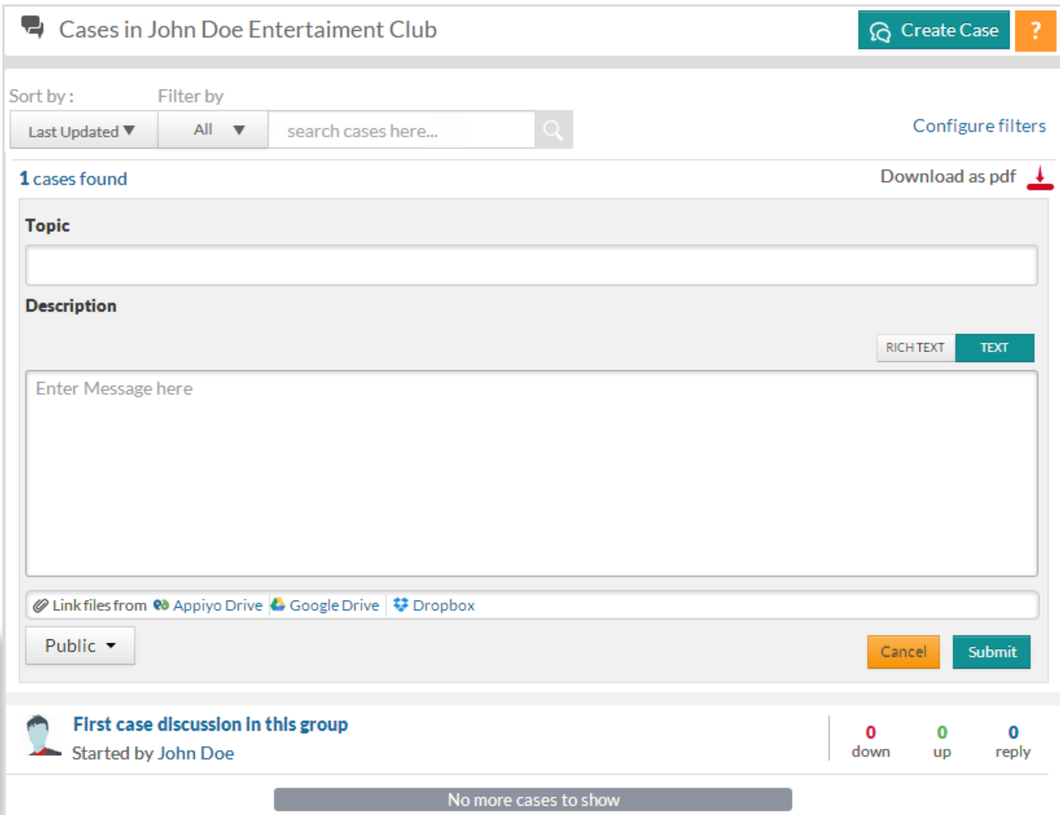
To **add your dislike to discussion** click the **icon** from the required discussion.

© Copyright 2014 Appiyo. All rights reserved.

Cases (Discussions)

Cases are useful to initiate and manage discussion around specific topics or issues. Typically they may be in the nature of a problem or a matter necessitating monitored attention. Appiyo provides a very useful feature for the case initiator to set priority **filters** uniquely for every case. These filters can be defined impromptu. There are no rigid type-cast master filters. How does this help? It enables every case to have a unique set of contextually meaningful filters depending on the nature of the issue. Different cases can be managed and acted upon differently making use of this nifty feature.

Navigation: Required Group → Cases tab → Create Case



To **create a case** for any group, click **Create Case** under **Cases** tab of the required group and enter necessary details under **Topic** and **Description**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

Set the type as **Public** or **Private.** You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit**.

To **notify** about a case to **all** group **members** (under case reply), click **Notify all**.

To **notify required members** (under Private cases), include the members list in the textbox near **Private** button.

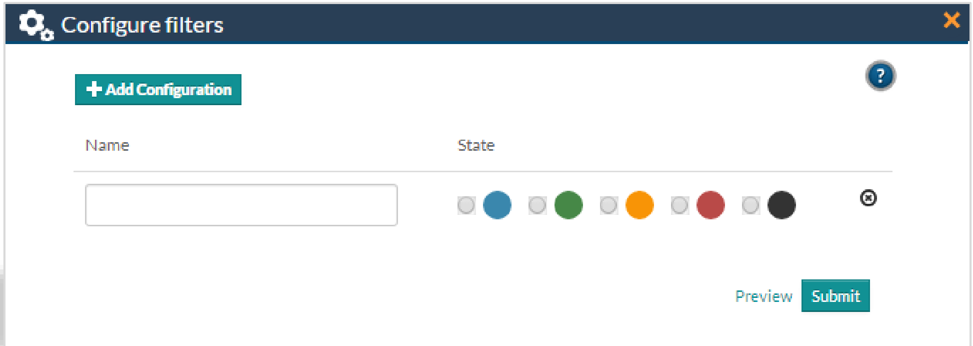
To **filter cases**, search for cases under **search cases here.** You can also search for cases by selecting either of the two options **Last Updated** or **Last Created** or by **defined case states.**

To **download** all the **cases as a pdf** file, click **Download as pdf**.

© Copyright 2014 Appiyo. All rights reserved.

Configure Case Filter

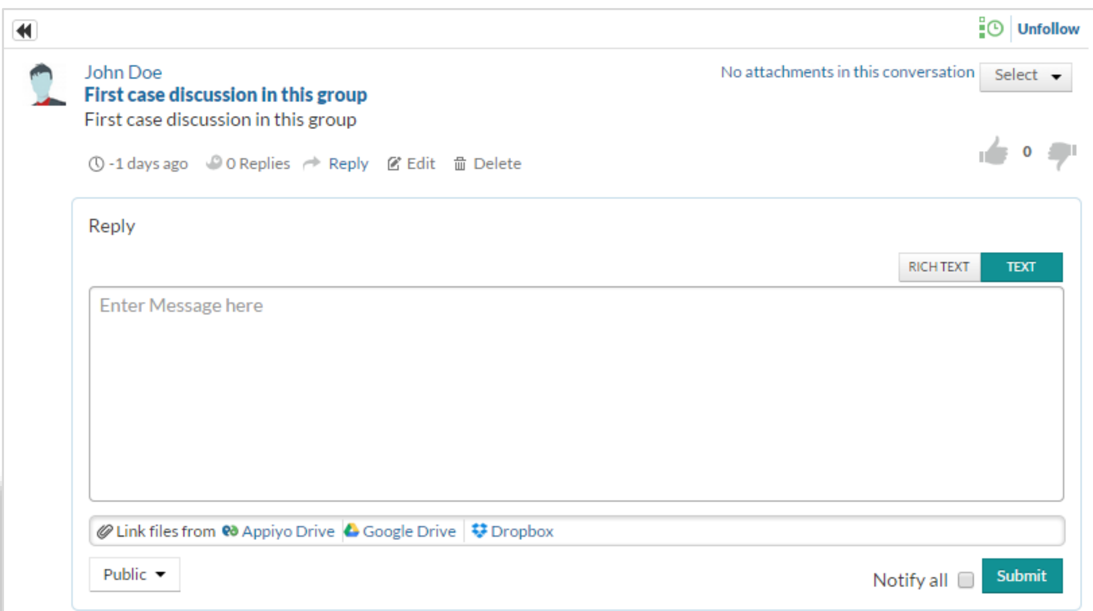
You can configure your case filter by defining your case states (error or assigned or fixed) using given colors.



To **add configure case filter**, click **Configure filters** and click **Add Configuration.**

Enter a name for the case state and assign a color for it from the given option buttons and click **Submit**.

To **edit case, delete case, follow/unfollow case, up/down vote** and **assign case configuration state**, click on the required case.



To **modify your case**, click **Edit** from the case (to be modified), make necessary changes and click **Update**.

To **delete your case**, click **Delete** from the case (to be deleted) and click **yes**.

To **get notifications about a case discussion,** click **Follow** of the required case.

To **stop getting notifications about a case discussion,** click **Unfollow** of the required case.

To **add your like to case discussion** click the **icon** from the required case.

To **add your dislike to case discussion** click the **icon** from the required case.

© Copyright 2014 Appiyo. All rights reserved.

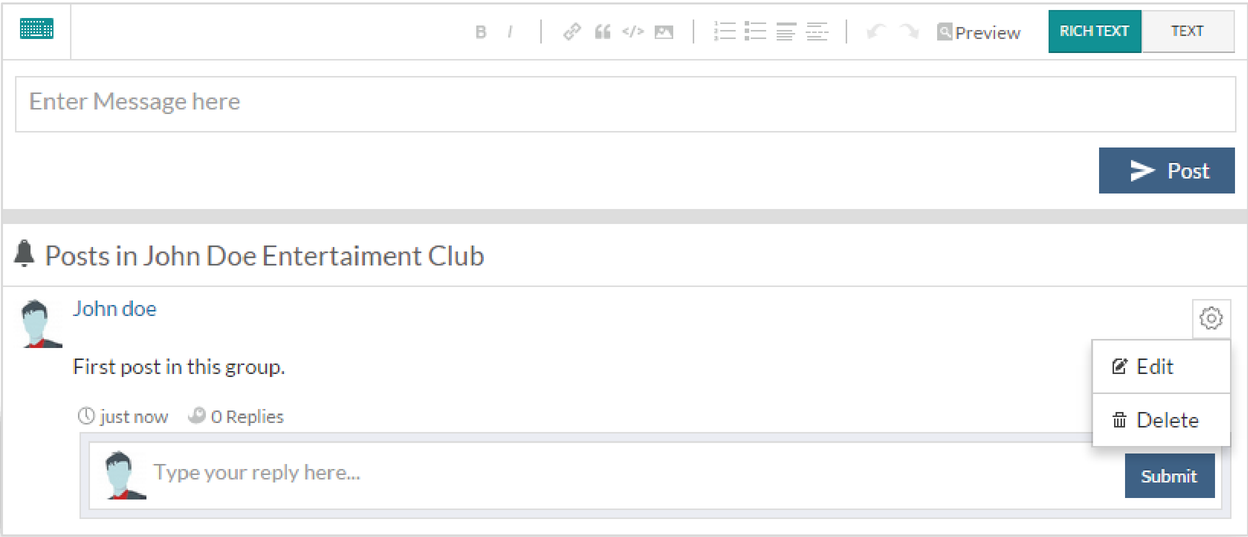
Assign Case Configuration State

Assign case configuration state allows you to assign added configure filter state to your case.

To **assign case configuration state**, click **Select** from the required case.

Posts

Navigation: Required Group → Posts tab



To post on any group, **Enter** your **Message** and click **Post**.

For **Message**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

To **reply to a post**, **Type your reply** and click **Submit**.

To **modify your post**, click **Edit** under **icon**, make necessary changes and click **submit**.

To **delete your post**, click **Delete** under **icon** and click **Yes**.

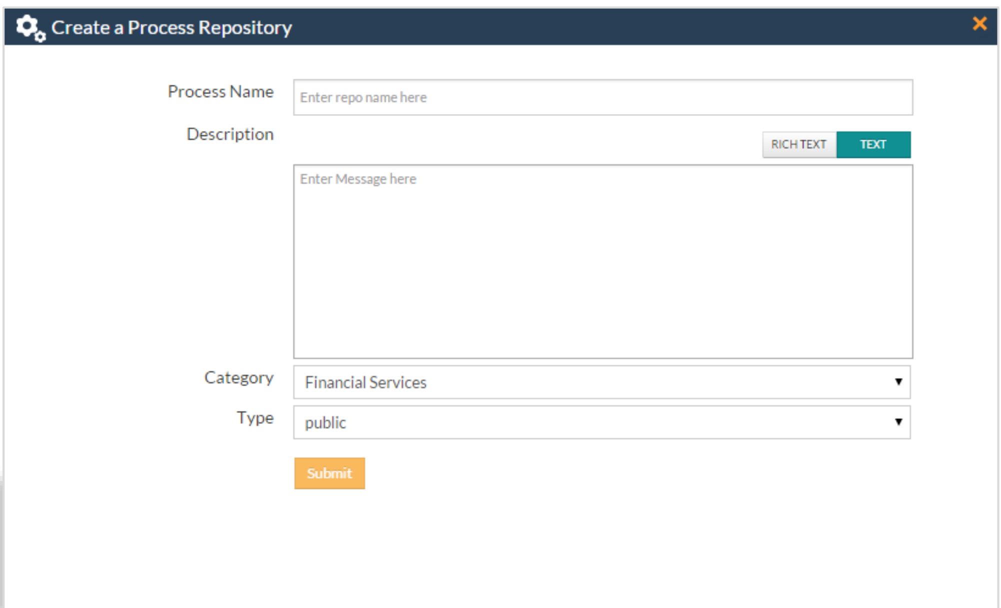
© Copyright 2014 Appiyo. All rights reserved.

Process Repository

- Here you find processes **you created** and processes **you have subscribed** to.
  - Appiyo refers to these as **process apps**.
  - Process apps are apps born out of and operated by Appiyo bpm.
  - **Repo subscribers** refer to people who have subscribed to your process apps. Their subscribership may influence initial opinions of prospective subscribers.
  - If your process app has more subscribers, many user comments and higher ratings, you have either created or, are on your way to creating a 'winner'.
  - If a process app description conveys clearly and concisely its purpose, it will draw attention. If it drifts, potential subscribers may not be attracted to it. So, it is important to provide a compelling '**pitch**' about your process app.
- You click on a process app to navigate to its next level of detail (also refer to Process).

Create Process Repository

Navigation: Process tab → Create a Repo



To create a process repository, click **Create a Repo** under **Process**.

Enter a name for the repos a description about the repos, also select a category from the drop-down list and set the type as **public, private** or **restricted**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

RICH TEXT – Includes formatting and hyperlink attachment options.

TEXT – Ignores formatting and hyperlink attachment options.

If type **public** is selected, all users in your zone can view the process.

If type **private** is selected, only subscribed persons can view the process. But it will be included in the process list.

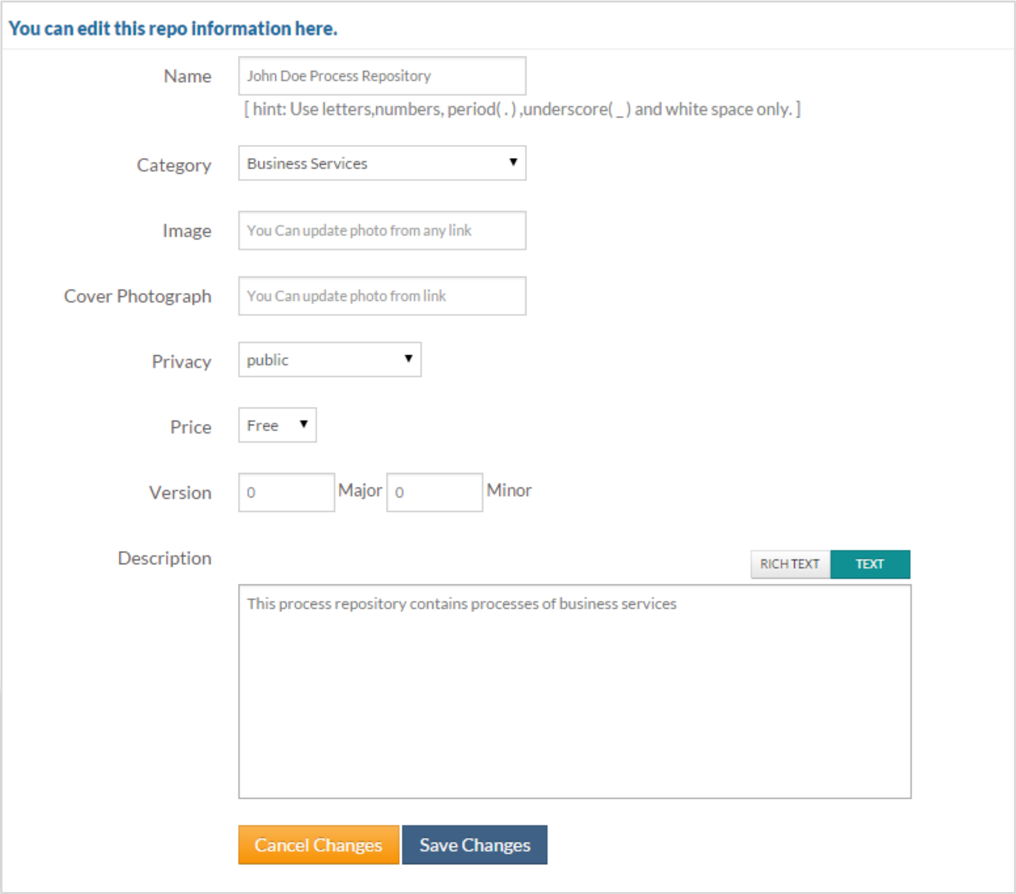
© Copyright 2014 Appiyo. All rights reserved.

If **restricted** is selected, only the owner and subscribed persons can view the process and it will not be displayed in the list of process.

Click **Submit**, the repos will be created and listed under Process.

Edit Repo allows you to modify your repos details.

Navigation: Required Process Repository → Description tab → Edit Repo (under repo cover photo)



To **modify your repos details** click **Edit Repo** (under repo cover photo) under **Description** tab of required process, make necessary changes and click **Save Changes.**

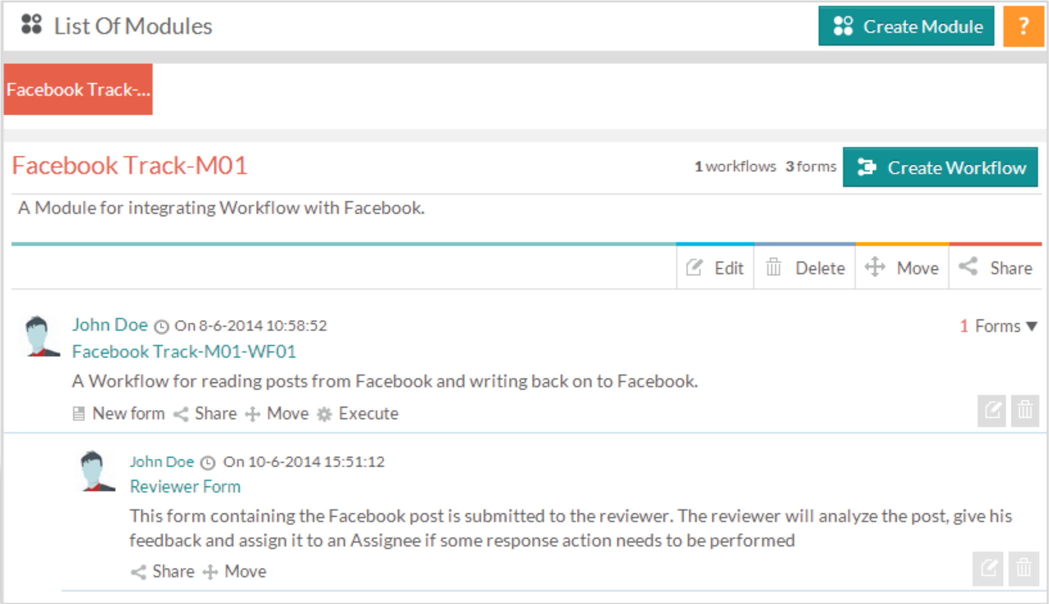
© Copyright 2014 Appiyo. All rights reserved.

Modules

A **module** refers to a logical collection of workflows.

Create a new module in your process. In modules, you can build your forms and workflows, execute and test their functionality.

Navigation: Required Process repository → Modules tab



For any created project, click **Create Module** button under **Modules** tab of required process, enter a name with description and click **Submit**.

To **modify your module**, click **Edit** of the module to be modified, make necessary changes and click **Submit**.

To **delete your module**, click **Delete** of the module to be deleted and click **Yes**.

© Copyright 2014 Appiyo. All rights reserved.

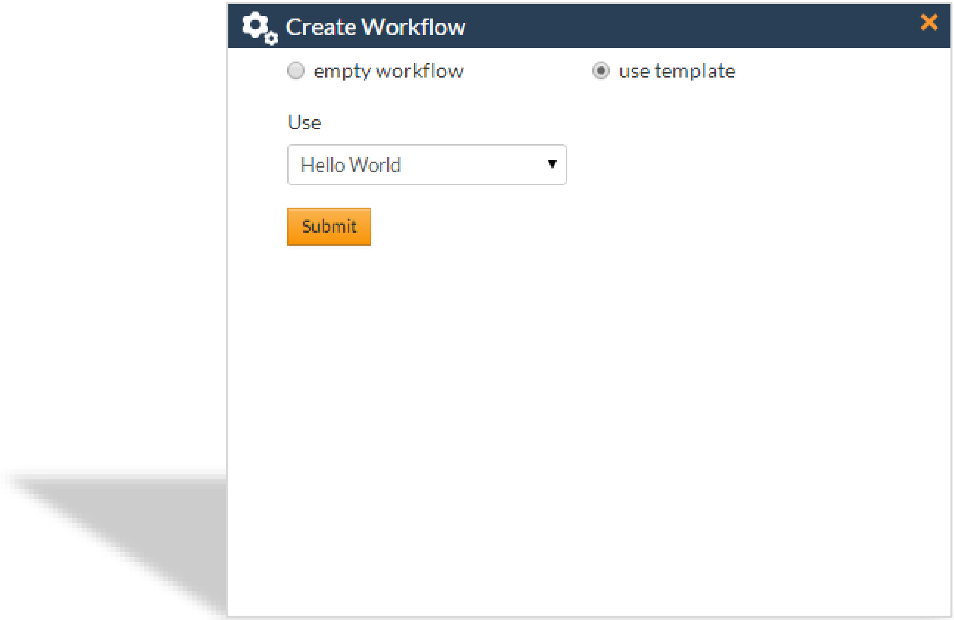
Workflows

- A workflow is a logical collection of sequenced tasks.
  - Within a workflow, you can create two meaningful separations.
  - A **pool** is one separation. It is a collection of tasks which can be called by more than one workflow. Pools of tasks enable parallel processing. You will use an "outer round rectangle" to denote a pool.
  - A **lane** within a pool is another separation. Typically this will be used to organize tasks belonging to a function or department. A lane within a pool can be spotted by a dividing line.
  - A disciplined use of modules, workflows, pools and lanes makes process execution and management very effective and less onerous.

Create a new Workflow

Here you can depict a diagrammatic representation of your project, deploy (compile) and execute them. This helps you analyze the process and enhance before their actual implementation. This reduces errors during implementation and ensures cost-effective testing.

To draw a workflow for a module created, click **Create Workflow** button and click either of the two option buttons - **empty workflow** or **use template.**



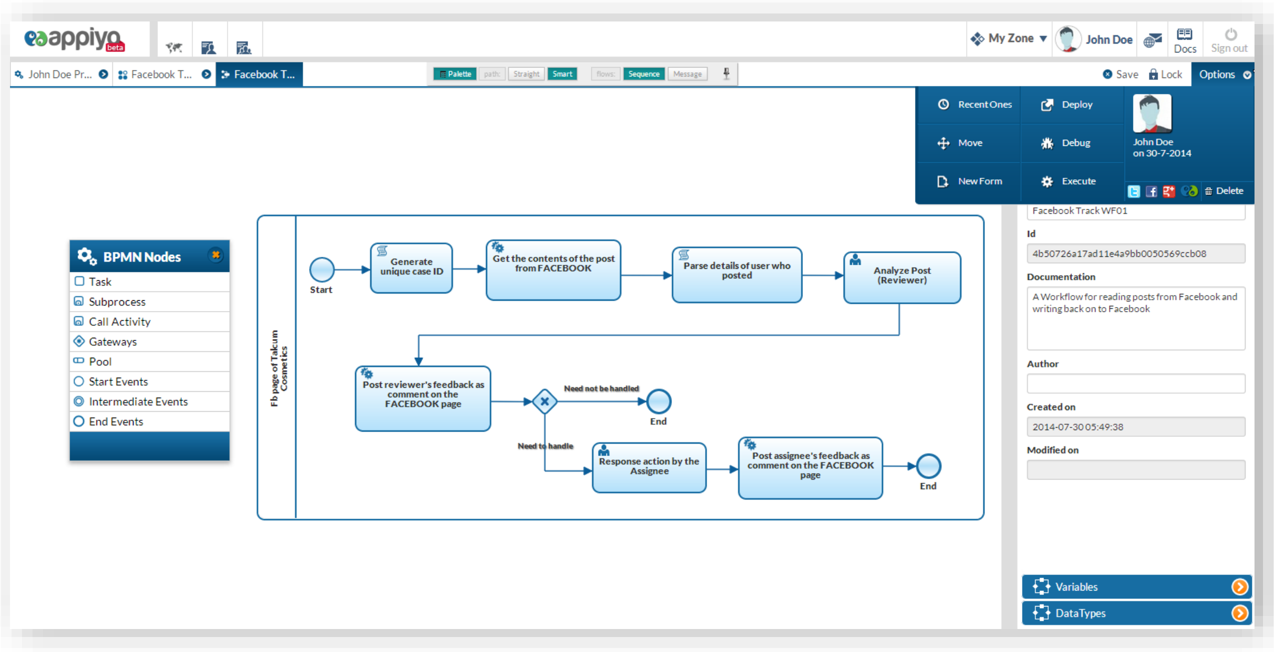
empty workflow- Add name and description for the workflow and click **Submit.**

use template - Select from pre- drawn workflow list (to use as a template) and click **Submit.**

© Copyright 2014 Appiyo. All rights reserved.

Draw a Workflow

Once created opens a work area to draw workflow using BPMN nodes.



Variables

Variable is a parameter that stores data required for process execution. They are used in process input/output mapping and data association. Types include: Primitive Data Types (text, character, number, decimal and boolean) and Appiyo Data Types (attachment - used for the form control Attachment).

To define variables:

1. Click on the required BPMN node to which you want to define variable
2. Click **Add Variable** under **Variables** on the right

Variable defined inside a pool is known as **process variable** whose scope is restricted only inside that pool. Variable defined inside an instance (like task, subprocess, call activity) is known as **instance/task variable** whose scope is restricted only inside that instance.

Types	Description
Complex Variables (DataTypes)	Collection of information (related with each other) in single table. Types include: **Default**, **Message**, **Error** and **Signal** (used by throw/catch nodes). For example, consider the variable ‘Employee’ that has variables like ‘name’ (text), ‘phone number’ (number) and ‘address’ (text) in it. ‘Employee’ is a complex variable since it has collection of information (like name, phone number) about an employee in it. A complex variable can also have inner complex variable. For example, the variable ‘address’ inside ‘Employee’ (complex variable) may have two variables within it as ‘address 1’ and ‘address 2’ (inner complex variables). **To define complex variables:** 1. Click **Add Datatype** under **DataTypes** on the right 2. Click under **Member Variables** to define variables inside the datatype

© Copyright 2014 Appiyo. All rights reserved.

Collection	It is an array of variables/complex variables. For example, the complex variable 'Employee' is a collection of information about one employee whereas 'Employees' is a collection of information about 'Employee' i.e. information about a list of employees. To **define collection**, select the checkbox **Collection** under **Add Variable**.

Note: A variable (name) can hold only one value (MyName) in it. In complex variables, each variable in it (name and phone) can hold one value (MyName and MyNumber). Collection (name) can hold multiple values (NameOne, NameTwo) in it.

While assigning defined variables as input/output for the process, variable should be included between '#' i.e. **#variablename#**

Icons

Icons	Definition	Configuration
	Used for defining the BPMN node type	1. Drag and drop the required BPMN node 2. Click the icon to define the required type
	Used for defining the node properties	1. Click the icon 2. Define the node properties
	Expression Builder Used for building expressions (for process input/output) by mapping defined variables	1. Click the icon 2. Click under **Map Variables** 3. Drag and drop required variables (from the variables list) and operators (like +,-) at right to build expressions **Example:** Variable1 + Variable2 = Variable3 Variable3 value is the addition of Variable1 value with Variable2 value.
	Data/Result Mapper Used for mapping one variable with another (refer to Database Access and Call Activity)	1. Click the icon 2. Drag and drop respective variables from the list at right (variable mapping) **Example:** Variable1 = Variable2 Variable1 defined inside another process is mapped with Variable2 defined inside current process.

© Copyright 2014 Appiyo. All rights reserved.

Start Events

Start event indicates the start of a process. Four types include:

Types	Symbols	Description
Start None		Start None event is triggered to start the process. One main process can have one Start None only whereas sub process can have one or more.
Message Start		A process receives message (sent from configured nodes like End Message/Send Task) from another process and starts the process execution. The message sent is thrown to only one process (i.e. receiver name is required). To receive the message, configure the message to be received under **Message Start Properties**. **Note:** The node that sends message (like End Message) should also be configured to point the process of the Message Start. **Possible Message Throw Nodes**: Send Task, Intermediate Throw Message, End Message.
Timer Start		Starts the process at the scheduled time or at scheduled time intervals (i.e. process repetition). To define the start time, configure it under **Start Timer Properties**. When scheduled time (**Once**) is defined, the process is executed only once. When scheduled time interval (**Many**) is defined, the process execution is repeated at scheduled time interval.
Signal Start		Similar to Message Start event, a process catches the required signal (broadcasted by configured nodes like End Signal) and starts the process execution. But the signal is broadcasted to all processes instead of throwing to one (i.e. receiver name is not required). Required processes catches the signal and starts the process. To catch the signal, configure the signal to be caught under **Select Signal To Be Received** (node properties of Signal Start). **Note:** The node that throws signal (like End Signal) should also be configured to point the process of the signal start. **Possible Signal Throw Nodes:** End Signal, Intermediate Throw Signal.

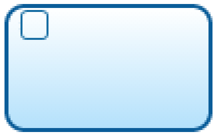
© Copyright 2014 Appiyo. All rights reserved.

Facebook Start		Facebook Start initiates process execution when subscribed events (new posts) occur in Facebook page. To start the Facebook process, define its properties under **Facebook Start Properties**. **Page Id** – Id of the Facebook page from which changes has to be tracked. **Output** **Item Name** – Items (Status, Post, Comment, Photo) which has to be tracked. **Trigger Identifier** – Variable (with Boolean value) assigned for each **Item** to trigger appropriate events. When an Item (Post) has changed (Boolean value - true), the event defined for it (Reply Post) will be triggered. **Value** – Variable (with Text value) assigned to get the unique id of the new post.

Note: Sub process cannot have Message Start, Timer Start and Signal Start events.

© Copyright 2014 Appiyo. All rights reserved.

Tasks (Activities)

Types	Symbols	Description
Task None		Used to evaluate expression (input/output mapping and data association).
Manual Task		Task that is performed by human without the intervention of process execution engine. **Example:** Consider a hardware engineer assigned to repair a system based on a customer's complaint. The repair work is done by the assigned person (without the intervention of process execution engine) manually.
Receive Task		Receives message from another source (sent from configured nodes like Send task) and proceeds with the process execution (similar to **Intermediate Message Catch Event**). It is used for conversation/discussion between processes. To receive the message, configure the message to be received under **Receive Task Properties**. **Correlation Id** – Defined variable that points the correct throw node. **Note:** The node that sends message (like send task) should also be configured to point the process of the receive task. Throw node should also have same **Correlation Id** as given in the Receive Task. **Possible Message Throw Nodes:** Send Task, Intermediate Throw Message, Message End.
Send Task		Sends message to message catchers and receivers (like intermediate message catcher and receive task) that waits for this message to proceed with their process execution To send message, configure the send task to point the process of the catchers/receivers under **Send Task Properties**. **Correlation Id** – Defined variable that points the correct catch node. **Note:** The node that receives message (like receive task) should also be configured to point the message thrown by the send task. Catch node should also have same **Correlation Id** as given in the Send Task. **Possible Message Catch Nodes:** Receive Task, Start Message, Intermediate Catch Message, Boundary Interrupting and NonInterrupting Message Events.

© Copyright 2014 Appiyo. All rights reserved.

Script Task		Executes the task automatically using scripts written in languages (currently supports Java/JavaScript) that can be parsed and executed by servers. Add required **Script** under the **Node Properties** of the script task on the right. **Example JavaScript** to get the input (Hello World) and show it as output *function execute(){* *var local;* *local = input;* *output = local ;* *}*
User Task		Waits for the human approval in between process execution and proceeds after getting the approval. Define **User Task Properties** **Add Potential Owners** – Assign the task to any single user (**User**) or to a group (**Group**) or to Participants (**Participant**). **Note:** To add a defined variable as **Id** value, click the **Process Variable** check box. To define user's function type, click the **Function** check box. **Form Id** – Enter the id of the form (refer to Form) to be shown to the assigned users. **Task Completion Type** – Variable assigned to get the task execution status (TASK_COMPLETED, TASK_OBSELETE, TASK_EXITED, TASK_ERROR). **Due Date** – Set the due date for the waiting time of user approval **Type** **Absolute** – Date deadline for user approval. **Relative** – Time deadline for user approval. **Is Skippable** – Skips the process. **Case Id** Task can also be monitored (under Instance of Monitor tab) by generating the case id for your user task. To generate the id, click on the pool and add the id under **CaseId** on the right. **Example: Leave Application Management** In leave application management process, an employee sends leave request to his manager which he may approve or reject and accordingly the reply for the request has to be sent back. Here the user task (assigned to manager) waits for the approval and continues the execution after getting it.

© Copyright 2014 Appiyo. All rights reserved.

Service Task		Tasks that consumes service and executed by the system. Web Service, HTTP Service, Mail Sender, Database Access, Alfresco Service, OpenERP, Salesforce Service, Facebook Service and Twitter Service are the supported Service Tasks.

Service Task Types

Types	Description
Web Service	Web Service helps data exchange between software systems (Service Requestor and Service Provider) over internet using Web Service Definition Language (WSDL) with Simple Object Access Protocol (SOAP).You may refer to the following steps to know how you can use Web Services from Appiyo. **Example:** *http://webservices.daehosting.com/services/TemperatureConversions.wso?WSDL* is a web service that can be executed to convert temperatures to required format. Define web service under **WebService Task Properties**. **WSDL location –** WSDL file location. **Load WSDL** loads the file from the specified location and shows the list of **Service**, **Port** and **Operation** to select. After selecting, map input and output variables **Request Payload –** Variable (with value) mapped to give input data. **Response Payload –** Variable mapped to get the output data.

© Copyright 2014 Appiyo. All rights reserved.

HTTP Service	Similar to Web Service HTTP service also helps data exchange between software systems (Service Requestor and Service Provider) over internet using Hyper Text Transfer Protocol. You may refer to the following steps to know how you can use HTTP Services from Appiyo. **Example:** *http://www.srh.noaa.gov/epz/?n=wxcalc_tempconvert* is an http service that can be executed to convert temperatures to required format. Define http service under **HTTPService Task Properties**. **Request** **Host –** Request url (http://www.srh.noaa.gov). **Path** – File location in the host (/epz/?n=wxcalc_tempconvert). **Version** – HTTP version used (1.0/1.1). **Method** – Request Method (GET, POST, PUT, DELETE). **Timeout** – Wait time for the response. **Content Type** – Request content type (Json, XML, HTML). **Headers** – Information (content length, content type) to be sent with the request. **Name** – Variable assigned for the header information (content_Type). **Value** – Value of the assigned variable. **Cookies** – User's cookie value (authentication token) for further access. **Name** – Variable assigned for authentication token. **Value** – Authentication token. **Request Payload Type** **Parameter** – Request sent as key-value pair (Json). **Name** – Key (temperatureType). **Value** – Value of the key (Celsius). **Raw Payload** – Raw request sent using XML, HTML. **Accept Response Type** – Required response type (Json, XML). **Accept Encoding Type** – Encoding type for the response. **Response** **Response Status** – Variable assigned to get the response status. **Content Type** – Variable assigned for the response content type. **Headers** – Information received with the response. **Name** – Variable assigned for the header information. **Value** – Value of the assigned variable. **Cookies** – User's cookie value (authentication token). **Name** – Variable assigned for authentication token. **Value** – Authentication token. **Response Body** – Variable assigned to get the response.

© Copyright 2014 Appiyo. All rights reserved.

Mail Sender	Appiyo is integrated with the internet standard for mail transmission - the Simple Mail Transfer Protocol (SMPT). Define mail sender under **Send Mail Properties**. **User Name –** Email id from which the mail has to be sent. **Password** – Password of the email id. **Host** – SMTP of the host. **Example SMTP:** **Yahoo:** smtp.mail.yahoo.com **Gmail:** smtp.gmail.com **Note:** For **Subject** or **Content** you can also add the defined variables that contains the actual value.
Database Access	Database stores and manages data using SQL. Database Access is for performing database related operations using the defined database connectivities, under Datasources. MySQL, PostgreSQL and SQL Server are the three types of SQL provided. Define database access under **Database Properties**. **System Datasource** Database connectivities defined under **Datasources** tab of the process. **Datasource Name –** Defined datasource name. **Selections –** Table fields (name) to be mapped with defined output variables (Name) to get their details (refer to Result Mapper). **Is Multi Row Insert** – Allows to insert multiple values inside multiple rows. **Query** – Query for the database operation (Create, Select, Insert, Update, Delete). **User Datasource** Database of the user. **Host** – Database host name (www.appiyo.com). **Port –** Database port number (8080). **Database** – Database name. **User Name** and **Password** – User id and password to connect to the database. **Selections** – Table fields (name) to be mapped with defined output variables (Name) to get their details (refer to Result Mapper). **Is Multi Row Insert** – Allows to insert multiple values inside multiple rows. **Query** – Query for the database operation (Create, Select, Insert, Update, Delete). **Example Create query (MySQL) for creating a table with birthday list** *create table BirthdayList(Name VARCHAR(50) NOT NULL, Birth_Date DATE NOT NULL);*

© Copyright 2014 Appiyo. All rights reserved.

Alfresco Service	Alfresco service is used to manage (upload or download) files. With alfresco user credentials files in it can be managed from Appiyo (similar to file management system). Define alfresco service under **Alfresco Properties**. **Host –** Host name (www.appiyo.com). **Port –** Port number of the host (8080). **User Name** and **Password** – User id and password to connect with alfresco. **Operation Upload Document** **Document Content** – File content to be uploaded. **Mime Type** –Mime type of the file (text/plain). **Document Path** – Location path to save the file in alfresco. **Document Name** – File name to save in alfresco. **Description** – Description about the file. **Document ID** – Defined variable that has file unique id as its value. **Operation Download Document** **Document Path (Path)** – Location of the file to be downloaded from alfresco. **Document Id (Id)** – Unique id of the file to be downloaded **Name** – File name to save the downloaded file **Mime Type** – Mime type of the file (text/plain). **Content** – Downloaded file content. **Operation Delete Document** **Document Id** – Unique id of the file to be downloaded **Operation Create Folder** **Folder Name** – Folder name to create in alfresco **Path** – Location path to create folder in alfresco **Description** – Description about the folder

© Copyright 2014 Appiyo. All rights reserved.

OpenERP	OpenERP is a business software that helps to manage your enterprise resources using Remote Procedure Call (RPC) (inter process communication) and XML. With OpenERP user credentials, these resources can be managed from Appiyo. Define OpenERP under **OpenERP Properties**. **Host** – Host name (www.appiyo.com). **Port** – Port number of the host (8080). **User Name** and **Password** – User id and password to connect with OpenERP. **Database** – Database name. **OpenERPmodel** – Name of the OpenERP model (res.partner) to be accessed. **Operation** – **Create**, **Read**, **Update**, **Delete** and **Search** record. **Input (Create and Update)** Field names (name, phone) present under specified OpenERP model and/or variables (#Name#, #Ph#) assigned to give the input data for specified operation. **Input (Read, Update and Delete)** **Record ID** – Variable with the record's unique id. **Input (Search)** **Offset** – Sets start point for the search query. If Offset is 2, then starts the search from second page. **Limit** – Sets limit for the search query. If Limit is 2, then search for only two records. **Query** – Query for search operation. **Example Search query for searching name** ("name", "=", "Myname") **Output (Create and Search)** **Record ID –** Assigned variable to get the record id of the created/searched file. **Output (Read)** Field names (name, phone) present under specified OpenERP model and/or variables (#Name#, #Ph#) assigned to get the output data. **Output (Update and Delete)** **Status** – Assigned variable to get the status of the operation.

© Copyright 2014 Appiyo. All rights reserved.

Salesforce Service	Salesforce is a cloud based database and social enterprise software used for customer relationship management. Using Salesforce user credentials, data or applications on Salesforce cloud can be accessed from Appiyo using REST API. Define salesforce service under **Salesforce Properties**. **User Name** – Salesforce user id. **Password** – Password followed by Security Token. **Version** – Currently used Salesforce version. **Client Secret** – Client's password and authentication token. **Client Id** – Client's consumer key. **Operation** – **Create**, **Update**, **Delete**, **Read**, **Query** and **Search** record. **Input** **Object** Name of the Salesforce object (Account) to be accessed. **Data (Create** and **Update)** Field names (Name, AccountNumber) present under specified Salesforce object and variables (#name#, #accNo#) assigned to give the input data for specified operation. **Record Id (Update, Delete** and **Read)** Assigned variable that has created record's id as its value. **Fields To Be Read (Read)** Name of the fields (Name, AccountNumber) to be read from specified Salesforce object. **SOQL Query (Query)** Salesforce Object Query Language (SOQL) queries to perform required operations. **Example Select query to select names that starts with 's' from the field 'Name' of object 'Account'** *SELECT Name from Account where Name like 's%'* **SOSL Query (Search)** Salesforce Object Search Language (SOSL) queries to search for records without specifying their object or field name. **Example Search query to search for record 'Myrecord'** *find {Myrecord}* **Output –** Assigned variables to get the output data.

© Copyright 2014 Appiyo. All rights reserved.

Facebook Service	Facebook Service feature helps you establishing a connection with Facebook from Appiyo (under Datasources) and get the Facebook notifications and post back on the Fb page. Define Facebook service under **Facebook Properties** **Read posts from a page, Read comments of a post and Read replies to a comment** **Node Id** – Id of the Facebook page from which the post has to be read or id of the post to be read (refer to Get Page and Post Id from Facebook). **Since** and **Until** – Read posts posted between specified time period. **Note:** Specify the time period in system readable format. Use **epoch converter** to convert the time. **Limit** – Maximum number of posts to be read. **Output** – Fields (**id**) mapped with defined variables (**Id** with page id as its value). **Write post to a page, Write comment to a post and Write reply to a comment** **Node Id** – Id of the page or post to which the post has to be posted (refer to Get Page and Post Id from Facebook). **Input** – Field (**message**) mapped with its defined variable (**msg**) that has the post to be posted in it. **Output** – Variable assigned to get the output (post information).
Twitter Service	Twitter Service feature helps you establishing a connection with Twitter from Appiyo (under Datasources) and get required details (like Tweets, Users). Define Twitter service under **Twitter Properties**. **Resource** Name of the twitter resource. **Request** Request to get required details from given twitter resource. **Input** Field (**id**) mapped with its value. **Output** Variable assigned to get the output.

© Copyright 2014 Appiyo. All rights reserved.

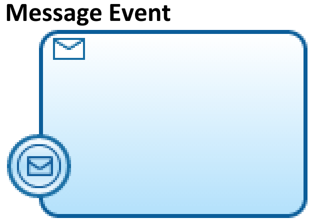
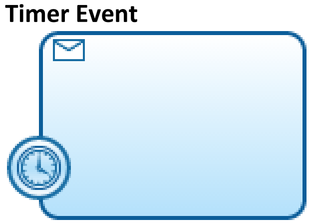
Boundary Events

Boundary events are used to set an alternate path for a task when the assigned path fails i.e. defined boundaries for task execution. Two types include: Boundary Interrupting and Boundary Non-Interrupting.

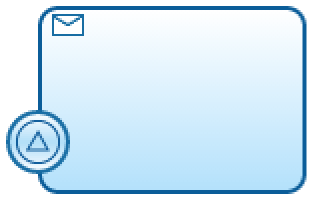
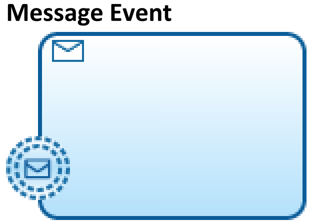
Boundary interrupting cancels the task execution and executes the attached events. Four types include: Message Event, Timer Event, Error Event and Signal Event.

Boundary non-interrupting executes the attached events without cancelling the task execution. Three types include: Message Event, Timer Event and Signal Event.

To add a boundary event to a node, right click on the node and click **Add Boundary Events**.

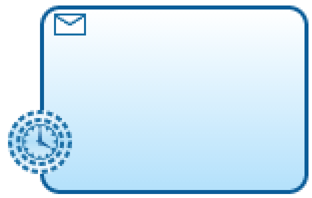
Types	Symbols	Description
Boundary Interrupting 	**Message Event** 	Receives information from another source (sent from configured nodes like Send Task). Once received interrupts the task execution and executes the attached boundary interrupting message event. To receive the message, configure the message to be received under **Message Boundary Interrupting Properties**. **Correlation Id** – Defined variable that points the correct throw node. **Note:** The node that sends message (like send task) should also be configured to point the process of the boundary interrupting message event. Throw node should also have same **Correlation Id** as given in the Message Event. **Possible Message Throw Nodes:** Send Task, Intermediate Throw Message, Message End.
	**Timer Event** 	Task waits for some information (message or signal) or some action (user approval or script specified actions) for specified time period. If the information is not received or the action is not completed (within specified time) task execution is interrupted and the attached boundary interrupting timer event is executed. To define the time period to wait, configure it under **Timer Boundary Interrupting Properties**. **Possible Nodes:** **Message Throw** Send Task, Intermediate Throw Message, Message End. **Signal Throw** Intermediate Throw Signal, End Signal. **Other Nodes** User Task, Script Task.

© Copyright 2014 Appiyo. All rights reserved.

	**Error Event** 	Receives the error message thrown by the configured **Error End**. Once received interrupts the task execution and executes the attached boundary interrupting error event. To receive the error message, include **Error Code** under **Node Properties** of boundary error event on the right. **Note:** Include same **Error Code** under **Node Properties** of both error end and boundary error event to throw and catch error message respectively.
	**Signal Event** 	Receives signal from another source (sent from configured nodes like intermediate signal throw). Once received interrupts the task execution and executes the attached boundary interrupting signal event. To receive signal, configure the signal to be received under **Select Signal To Be Received**. **Note:** The node that sends signal (like intermediate throw signal) should also be configured to point the process of the boundary interrupting signal event. **Possible Signal Throw Nodes:** Intermediate Throw Signal, End Signal.
Boundary Non-Interrupting 	**Message Event** 	Receives information from another source (sent from configured nodes like send task) and starts the execution of attached boundary non-interrupting message event. But the task execution will not be cancelled instead will continue execution once its requirements are met. To receive the message, configure the message to be received under **Message Boundary NonInterrupting Properties**. **Correlation Id** – Defined variable that points the correct throw node. **Note:** The node that sends message (like send task) should also be configured to point the process of the boundary noninterrupting message event. Throw node should also have same **Correlation Id** as given in the Message Event. **Possible Message Throw Nodes:** Send Task, Intermediate Throw Message, Message End.

© Copyright 2014 Appiyo. All rights reserved.

Timer Event



Task waits for some information (message or signal) or some action (user approval or script specified actions) for specified time period. If the information is not received or the action is not completed (within specified time) the attached boundary non interrupting timer event is executed. But the task execution will also continue once its requirements are met.

To define the time period to wait, configure it under **Timer Boundary NonInterrupting Properties**.

Possible Nodes:

Message Throw

Send Task, Intermediate Throw Message, Message End.

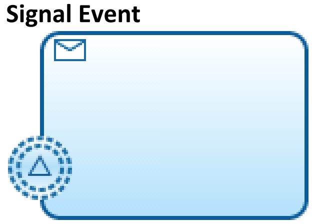
Signal Throw

Intermediate Throw Signal, End Signal.

Other Nodes

User Task, Script Task.

Signal Event



Receives signal from another source (sent from configured nodes like intermediate signal throw). Once received executes the attached boundary non-interrupting signal event. But the task execution will not be cancelled instead will continue execution once its requirements are met.

To receive signal, configure the signal to be received under **Select Signal To Be Received** (node properties of Signal Event).

Note: The node that sends signal (like intermediate throw signal) should also be configured to point the process of the boundary noninterrupting signal event.

Possible Signal Throw Nodes:

Intermediate Throw Signal, End Signal.

A task itself can act as a boundary to the attached events i.e. if the task is executed, attached events will be cancelled.

© Copyright 2014 Appiyo. All rights reserved.

Gateways

Gateways are the decision maker for the process flow path. When one node is merged to single gateway, it is called as **Merging Gateway.** When multiple nodes are merged to single gateway, it is called as **Forking Gateway.** Four supported types include Exclusive Gateways, Inclusive Gateways, Parallel Gateways and Event Based Gateways.

Types	Symbols	Description
Exclusive Gateways		When a process has two or more conditional (either or) flows, exclusive gateways are used to direct the execution towards one flow that satisfies the specified condition. **Example :** In leave application management process, an employee sends leave request to his manager which he may approve or reject (either or condition). Here exclusive gateway is used to make the flow decision based on manager's action. If the manager approves the leave request, the gateway allows the flow of sending reply to the employee as "Approved". If the manager rejects the request, the gateway allows the flow of sending reply as "Rejected". **Note:** Exclusive gateway allows only one flow execution satisfying the condition.
Inclusive Gateways		When a process has two or more conditional flows, inclusive gateways directs the execution towards the flows (more than one) that satisfies the specified condition. **Example :** In leave application management process, an employee's leave request may be approved or rejected (two conditions) by the manager. If the first condition is met i.e. leave request is approved, it has to be informed to the employee as well as to the accounts department. Hence here inclusive gateway is used to allow the execution of both the flows (informing employee and accounts department) satisfying the first condition. **Note:** In above example, parallel gateway can also be used to execute both the flows (informing employee and accounts department) in parallel.
Parallel Gateways		Parallel gateway allows execution of two or more unconditional flows in parallel and completes all before executing the outgoing flow. **Example:** In leave application management process, when an employee's leave request is approved by the manager it has to be informed to both employee and accounts department. Hence here parallel gateway is used to execute both the flows in parallel.

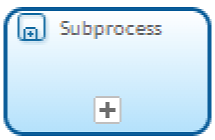
© Copyright 2014 Appiyo. All rights reserved.

Event Based Gateways		Event based gateways directs the flow based on the events (specified for the flow) occurred. **Example:** In customer order fulfillment process, after sending quotation to the customer the order has to be handled based on the customer's action (two events). Event based gateway directs the flow towards order delivery when the customer confirms the order (first event) else directs the flow towards sending reminders to the customer (second event).

Subprocess

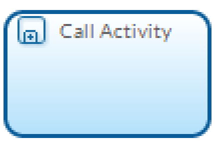
Subprocesses are the embedded processes that can have process inside them. The process (defined inside subprocess) scope is restricted only inside that node. This helps to hide the complexity of the process.

Click to define a subprocess.



Call Activity

Call Activity can define process variables that can be reused in another process i.e. this defined process variables can be called by another process. But their scope is restricted only inside that project (i.e. process repository).



To reuse the process variables (defined inside process of call activity) in another process:

1. Configure the target process under **CallActivity Properties**
2. Map the target process's variables with the respective variables defined inside call activity using **Data Mapper** ( ).

© Copyright 2014 Appiyo. All rights reserved.

Intermediate Events

Waits for some information (message or signal) or some action (user approval or script specified actions) in between process execution. It is mostly used as attached events. Two types include: Catching Intermediate and Throwing Intermediate.

Catching Intermediate

Catches required message or signal (thrown by Throw Nodes) and executes the process. Four types include:

Types	Symbols	Description
Message		Receives message from another source (sent from configured nodes like intermediate throw message) and proceeds with the process execution (similar to **Receive Task**). To receive the message, configure the message to be received under **Intermediate Catch Message Properties**. **Correlation Id** – Defined variable that points the correct throw node. **Note**: The node that sends message (like intermediate throw message) should also be configured to point the process of the intermediate catch message. Throw node should also have same **Correlation Id** as given in the Message. **Possible Message Throw Nodes:** Intermediate Throw Message, Send Task, Message End.
Timer		Waits for some information (message or signal) or some action (user approval or script specified actions) in between process execution for specified time period (currently supports ISO format). To define the time period to wait, configure it under **Intermediate Catch Timer Properties**. **Possible Nodes:** **Message Throw** Send Task, Intermediate Throw Message, Message End. **Signal Throw** Intermediate Throw Signal, End Signal. **Other Nodes** User Task, Script Task.

© Copyright 2014 Appiyo. All rights reserved.

Link (Catch and Throw)		To reduce complexity in representation, tasks pointing to same event will be connected using Link events (without using flow sequence) i.e. all the tasks pointing to same event is connected to **Link Throw** (whose target is set as Link Catch) and the **Link Catch** is connected to the event pointed by the tasks. For example, consider six tasks that have to point to end event. All six tasks will be connected to Link Throw event whose target is set as Link Catch event connected to the end event. In Link events, Link Throw supports only incoming flow and Link Catch supports only outgoing flow. To set the target, click on the **Link Throw** and add the node **Id** (shown on right) of the required **Link Catch** as **Target** under **Node Properties** shown on right.
Signal		Receives signal from another source (sent from configured nodes like intermediate throw signal) and proceeds with the process execution. To receive the signal, configure the signal to be received under **Select Signal To be Received** (node properties of intermediate catch signal). **Note**: The node that sends signal (like intermediate throw signal) should also be configured to point the process of the intermediate catch signal. **Possible Signal Throw Nodes:** Intermediate Throw Signal, Signal End.

© Copyright 2014 Appiyo. All rights reserved.

Throwing Intermediate

Throws message or signal to the configured receivers (like Catch Intermediates). Four types include:

Types	Symbols	Description
None		Representation of intermediate event. Used to indicate the change in state of the process.
Message		Sends message (similar to **Send Task**) to message catchers and receivers (like intermediate message catcher and receive task) that waits for this message to proceed with their process execution. To send message, configure the intermediate throw message to point the process of the catchers/receivers under **Intermediate Message Throw Properties**. **Correlation Id** – Defined variable that points to correct catch node. **Note:** The node that receives message (like intermediate catch message) should also be configured to point the message thrown by the intermediate throw message. Catch node should also have same **Correlation Id** as given in the Message. **Possible Message Catch Nodes:** Intermediate Catch Message, Receive task, Message Start.
Link		Refer to **Link (Catch and Throw)**.
Signal		Sends signal to signal catchers (like intermediate signal catcher) that waits for this signal to proceed with their process execution. To send signal, configure the signal to be sent under **Select Signal To Be Received** (node properties of Intermediate Throw Signal). **Possible Signal Catch Nodes:** Intermediate Catch Signal, Signal Start. **Note:** The node that catches signal (like intermediate catch signal) should also be configured to point the signal thrown by the intermediate throw signal.

© Copyright 2014 Appiyo. All rights reserved.

End Events

End events indicate the end of a process. End events always throw events. Four types include:

Types	Symbols	Description
End None		End None is triggered to end the process.
End Message		Sends message to message catchers and receivers (like message start and intermediate message catcher) that waits for this message to proceed with their process execution. To send message, configure the message to be sent under **End Message Properties**. **Correlation Id** – Defined variable that points to correct catch node. **Note:** The node that receives message (like start message) should also be configured to point the message thrown by the end message. Catch node should also have same **Correlation Id** as given in the End Message. **Possible Message Catch Nodes:** Message Start, Intermediate Catch Message and Receive task.
Error End		Sends error message to the configured **Boundary Interrupting Error Event** that waits for this error message to proceed with their process execution. This helps to take required action (specified by the error event) when an error is caught during process execution. To send the error message, include **Error Code** under **Node Properties** of error end on the right. **Note:** Include same **Error Code** under **Node Properties** of both error end and boundary error event to throw and catch error message respectively.
End Signal		Similar to End Message event, End Signal throws signal. But the signal is broadcasted to all processes instead of throwing to one (i.e. receiver name is not required). Required processes catches the signal and ends the process. To throw the signal, configure the signal to be thrown under the node properties of End Signal. **Note:** The node that catches signal (like start signal) should also be configured to point the signal thrown by the end signal node. **Possible Signal Catch Nodes:** Signal Start, Intermediate Catch Signal, Boundary Interrupting and Non Interrupting Signal Events.

© Copyright 2014 Appiyo. All rights reserved.

Pool

A pool is considered as one process within an organization. Process participants (like tasks, events) are represented inside pool. Number of pools is equal to the number of processes.

Lane

Subdivision of pool to represent the process flow within different sections of an organization.

Example

For employee's leave application, the department manager has to interact with the HR and the HR has to reply the department manager. The manager has to send back HR's reply to the employee and to the accounts department.

Though above example is the execution of one process, three lanes will be used to represent each department (manager, HR and accounts department).

© Copyright 2014 Appiyo. All rights reserved.

Looping

Used to define the execution behavior of process instances (like tasks, subprocess and call activity).

To define the loop type (three types) and their properties, click **Loop Properties** (of required instance) on the right.

Types	Symbols	Description
Standard		One instance is executed multiple times. They are mostly used for sequential executions like calculations. **Standard loop properties** **Test before** True = First checks the defined condition and executes only if the condition is satisfied (While loop). False = Executes the process at least once before checking the condition (do While loop). **Loop condition** Condition defined for the process execution. **Loop maximum** Number of times the loop has to be executed. **Note:** When both Loop condition and Loop maximum is defined, by default only Loop condition will be executed.
MultiInstance Parallel		All the instances are executed in parallel. They are independent of their parent instance i.e. execution of first instance will not affect the execution of its subsequent instances. **MultiInstance Parallel loop properties** **Loop cardinality** Number of times the loop has to be executed. **Loop data input** Process variables (i.e. defined inside a pool) to give the input data whose scope is restricted only inside that process (i.e. pool). **Loop data output** Defined process variables to get the output data. **Input data item** Instance variables (i.e. defined inside process instance) to give the input data whose scope is restricted only inside that instance (i.e. task/subprocess/call activity). **Output data item** Defined instance variables to get the output data. **Example:** Sending mail to all employees in an organization is executed in parallel (also refer to addition of variables example).

© Copyright 2014 Appiyo. All rights reserved.

MultiInstance Sequentials

Multiple instances are executed multiple times while each and every instance is also sequentially executed i.e. execution of first instance will affect the execution of its subsequent instances. They are similar to standard looping.

MultiInstance Sequentials loop properties

Loop cardinality

Number of times the loop has to be executed.

Loop data input

Process variables (i.e. defined inside a pool) to give the input data whose scope is restricted only inside that process (i.e. pool).

Loop data output

Defined process variables to get the output data.

Input data item

Instance variables (i.e. defined inside process instance) to give the input data whose scope is restricted only inside that instance (i.e. task/subprocess/call activity).

Output data item

Defined instance variables to get the output data.

Example:

Consider a process of addition of variables a = a + a and b = b + b where a = 1, b = 2 and loop cardinality is 2.

Execution of first instance (a and b in parallel)

a = a + a = 1 + 1 = 2

b = b + b = 2 + 2 = 4

Execution of second instance (a and b in parallel)

a = 2 (output of first instance)

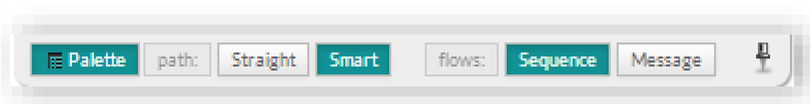
a + a = 2 + 2 = 4

b = 4 (output of first instance)

b + b = 4 + 4 = 8

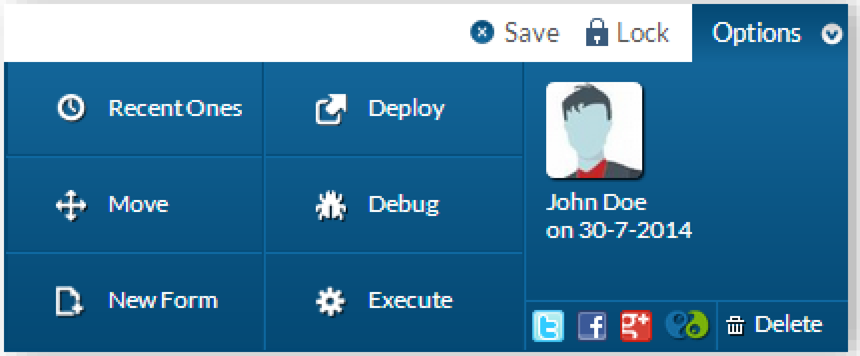
Here execution of second instance is affected (getting input) by first instance. But in MultiInstance Parallel execution of first instance will not affect the second instance i.e. the value of a (=1) and b (=2) will be same for all the instances.

Flow Sequence



Flow Sequence line is used to show that one resource is interacting with the other for obtaining information.

© Copyright 2014 Appiyo. All rights reserved.



Save Workflow

Click **Save** to effect the changes in your new or modified workflow. (The changes made will be sent as a request with their details to REST API to save the modifications).

Lock and Unlock Workflow

Click **Lock** to prevent other users from modifying the workflow you are working on. **Unlock** will allow the users in your zone to modify the workflow.

Deploy Workflow

Compile your workflow. This displays the number of errors (if any) in your workflow with their details.

To **compile workflow**, click **Deploy** under **Options.**

Debug Workflow

Debug allows you to execute each and every node on your workflow one by one.

To **debug workflow** (after successful deployment), click **Debug** under **Options** and select the form to be used from the list after creating a form.

Click **Open** and **Submit**.

Click **Stop over** (to stop at each node after their debug) or **Run** (complete debug).

Execute Workflow

Execute your workflow and check whether it gives back the appropriate result. You can execute it only after its successful deployment which means your workflow should not have any error.

To execute workflow, click **Execute** under **Options** and select the process to be executed from the list.

Select the Form to be used from the list after creating a form.

Click **Open**, enter values and click **Submit**; the result is displayed.

Move Workflow (Module to Module)

Move Workflow allows you to move your workflow from one module to another.

To **move a workflow**, click **Move** of required workflow and select the target module name to which the workflow is to be moved from the drop down list and click **Submit**.

© Copyright 2014 Appiyo. All rights reserved.

Edit Workflow

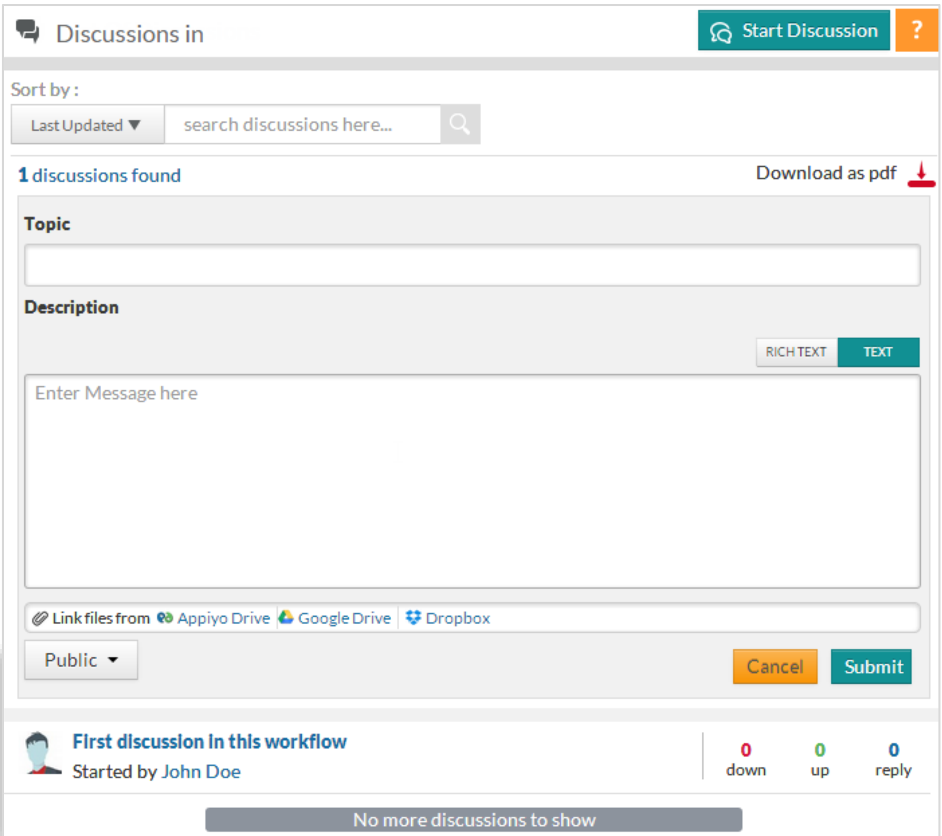
To **modify your workflow**, click the **icon** from the workflow (to be modified) under **Modules** tab, make necessary changes and click **Submit**.

Delete Workflow

To **delete your workflow**, click the **icon** from the workflow (to be deleted) and click **yes**.

Discussion

Navigation: Required Workflow → Discussion tab (on the right) → Start Discussion



To **start a discussion** for any workflow, click **Start Discussion** under **Discussion** tab on the right and enter necessary details under **Topic** and **Description**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

Set the type as **Public** or **Private.** You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit**.

To **notify** about a discussion to **all** process repository **members** (under discussion reply), click **Notify all**.

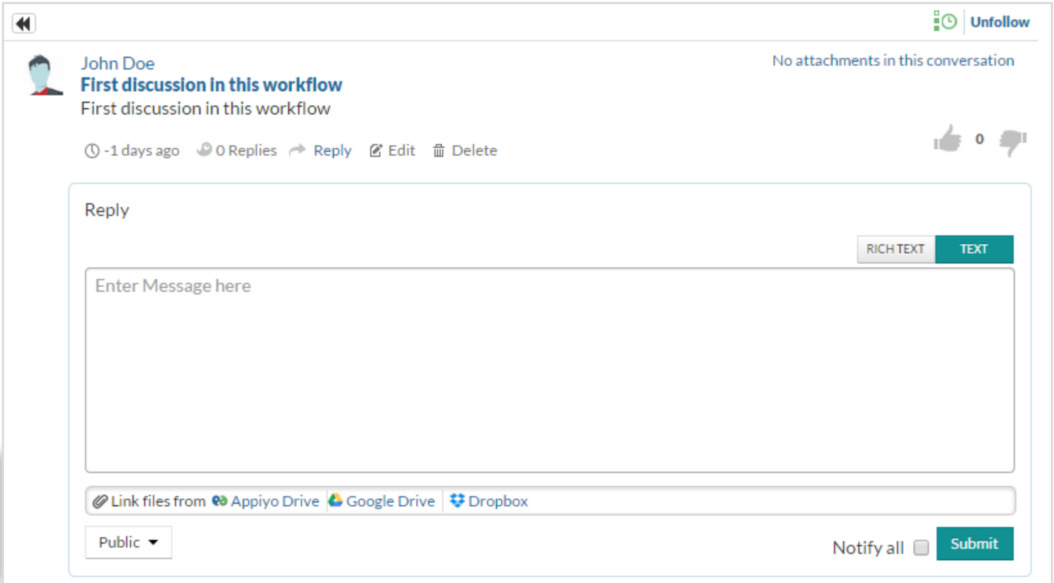
To **notify required members** (under Private discussions), include the members list in the textbox near **Private** button.

To **filter discussions**, search for discussions under **search discussions here**. You can also search for discussions by selecting either of the two options **Last Updated** or **Last Created**.

To **download** all the **discussions as a pdf** file, click **Download as pdf**.

© Copyright 2014 Appiyo. All rights reserved.

To **edit discussion, delete discussion, follow/unfollow discussion** and **up/down vote**, click on the required discussion.



To **modify your discussion**, click **Edit** from the discussion (to be modified), make necessary changes and click **Update**.

To **delete your discussion**, click **Delete** from the discussion (to be deleted) and click **yes**.

To **get notifications about discussions,** click **Follow** of the required discussion.

To **stop getting notifications about discussions,** click **Unfollow** of the required discussion.

To **add your like to discussion** click the **icon** from the required discussion.

To **add your dislike to discussion** click the **icon** from the required discussion.

© Copyright 2014 Appiyo. All rights reserved.

Workflow Example

The example workflow below uses Facebook Service tasks to connect from Appiyo with the Facebook page of a company called Talcum Cosmetics Pvt. Ltd. This will enable Appiyo to track the posts on the Facebook page posted by the customers of Talcum Cosmetics and suggest suitable solutions.

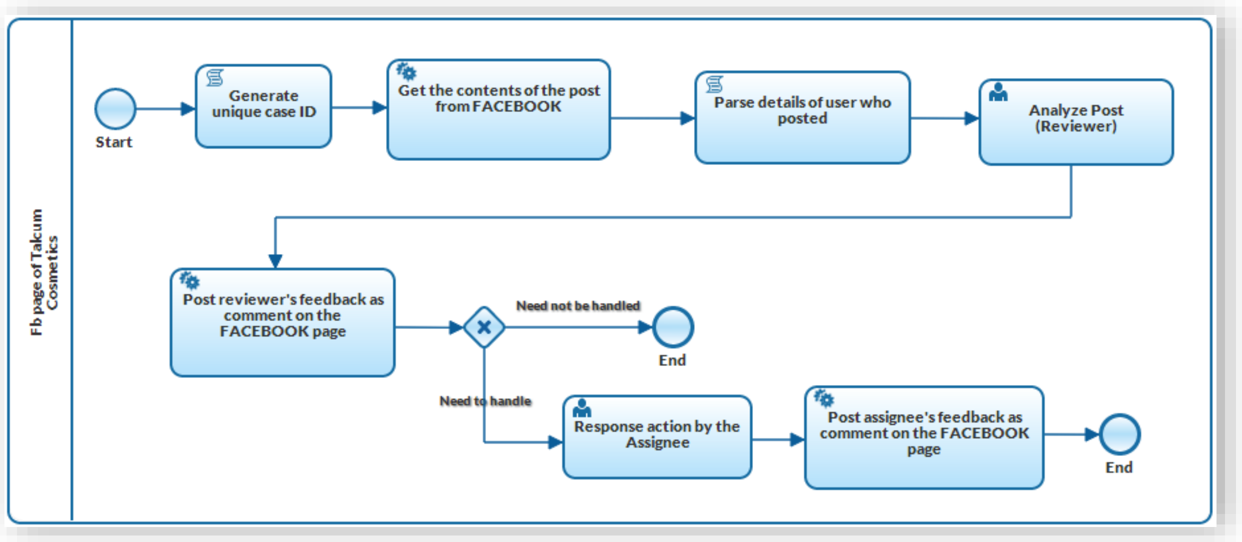
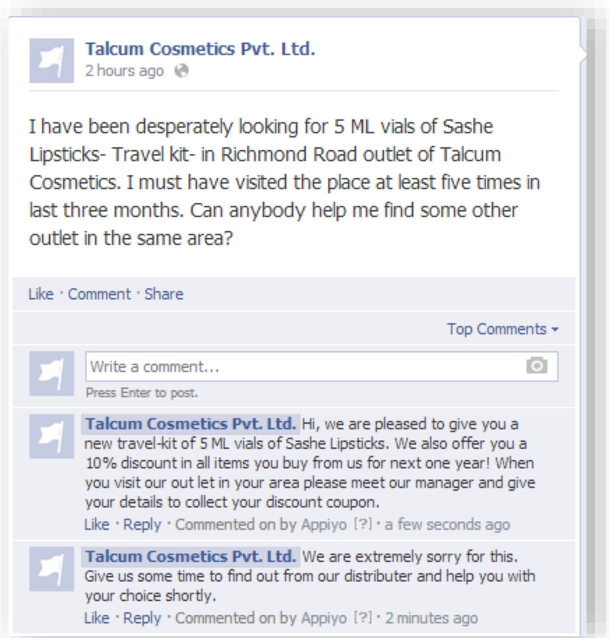


Figure 1 – Workflow for Talcum Cosmetics Facebook Track

Output

Reply (from Appiyo) by the reviewer and the assignee to the tracked customer's post will be reflected on Talcum Cosmetics Facebook page.

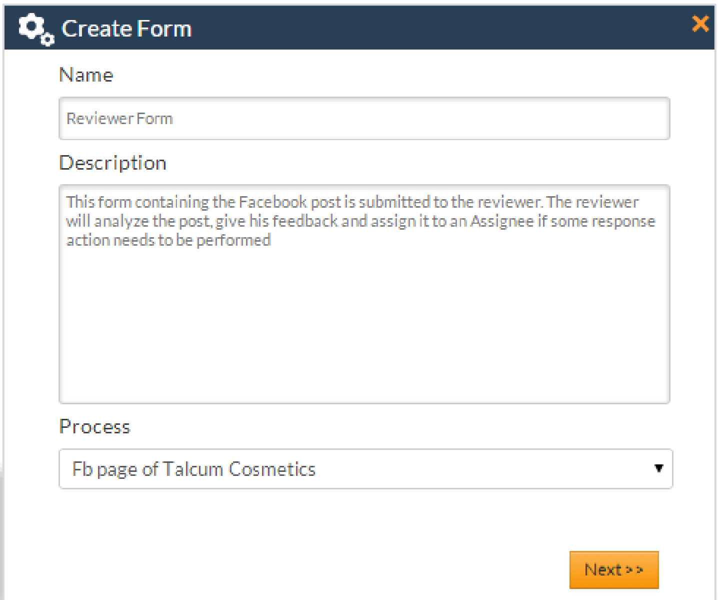


© Copyright 2014 Appiyo. All rights reserved.

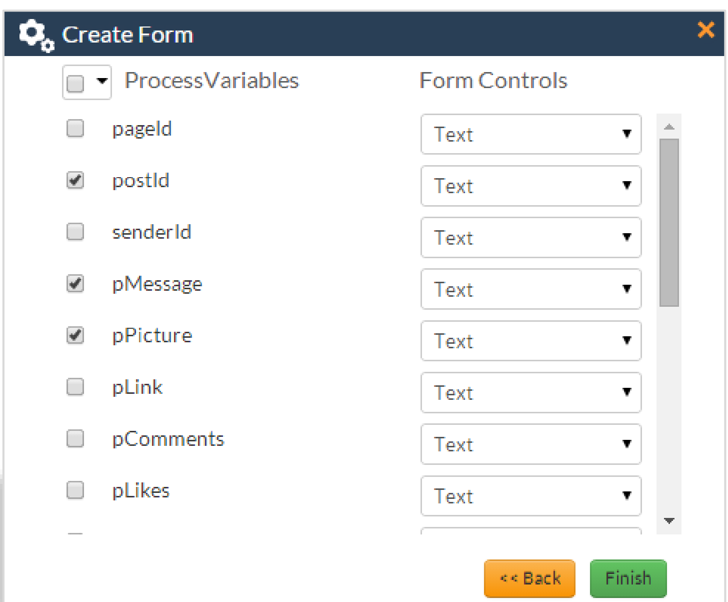
Forms

Create a new form for your workflow to give the input for workflow execution and also to show the resultant output after the execution. So you need to create a form for your workflow before you execute it.

To create a new form, click **New Form** under **Options**. If not created a form before workflow execution, system prompts message to create one.



Enter a name, description and click **Next**.



Select the check box **Process Variables** (to include all the defined variables) or select required check boxes.

Select the **Form Controls** from the drop down list and click **Finish**.

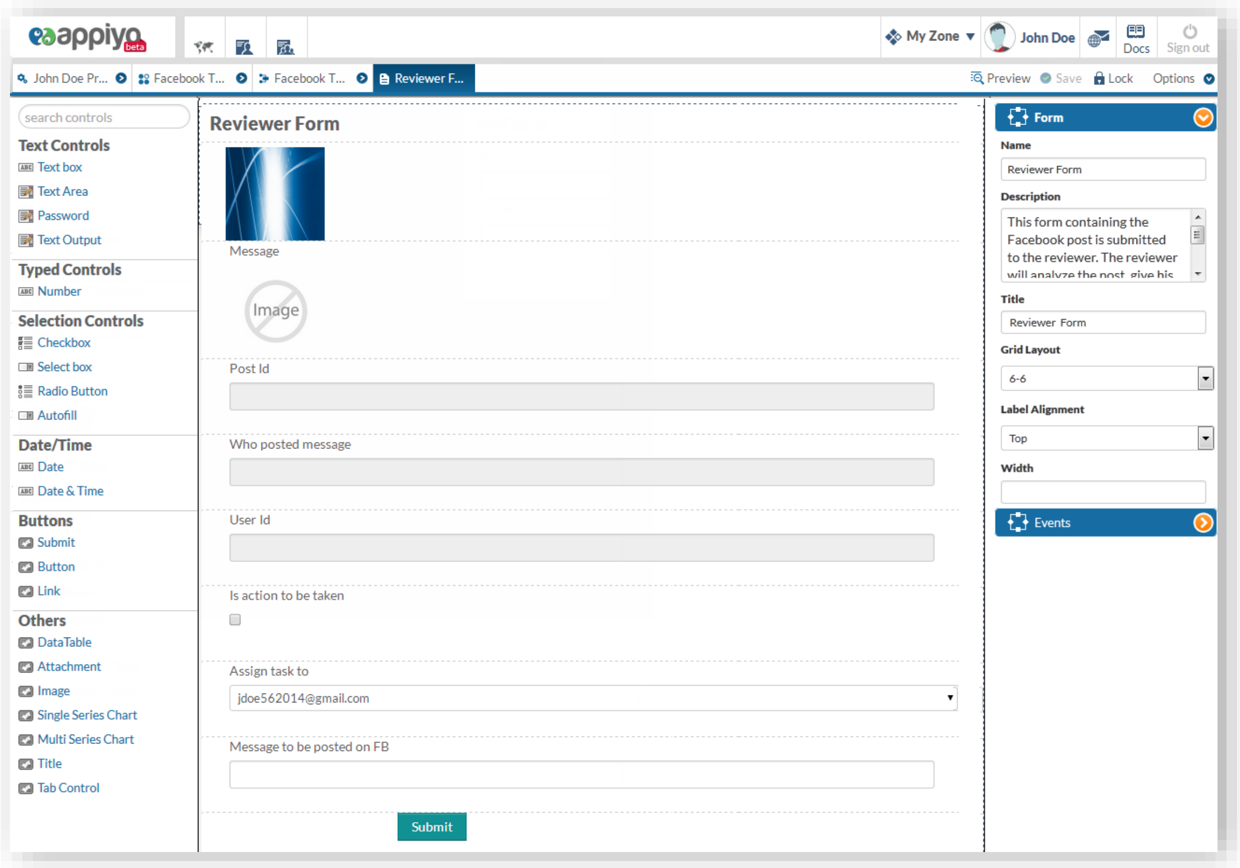
© Copyright 2014 Appiyo. All rights reserved.

To **modify your form details**, click the **icon** from the form (to be modified) under **Modules** tab, make necessary changes and click **Submit**.

To **delete your form**, click the **icon** from the form (to be deleted) and click **Yes**.

To **modify your form design**, click on the required form under **Modules** tab, make necessary changes (drag and drop required form controls from left) and click **Save.**

Click **Lock** to prevent other users from modifying the form you are working on. **Unlock** will allow the users in your zone to modify the form.



Form Controls

Types

Text Controls

Used for entering text values. All text controls should be mapped to text variables in workflow.

Text box	Only single line text is allowed.
Text Area	Multiple line text is allowed.
Password	Shows encrypted characters.
Text Output	Used to display text (like message) as output.

© Copyright 2014 Appiyo. All rights reserved.

Typed Controls

Number	Used for entering numbers. Number should be mapped with respective number variables in workflow.

Selection Controls

Used for selecting any option.

Checkbox	Used for selecting the Boolean option (true/false) that is mapped with respective Boolean variables in workflow.
Select box	Used for selecting options from the drop down list.
Radio Button	Used for selecting one option button from many.
Autofill	Used for selecting the option that is shown when a value (if it matches with the defined options) is entered.

Date/Time

Date	Used for selecting the date.
Date & Time	Used for selecting date and time.

Buttons

Submit	On click will call the workflow to be executed and moves on to the next page/form. Example:- In online purchase, clicking 'Pay' after adding items to the cart will proceed to the payment gateway i.e. moves on to the next page/form.
Button	On click will call the workflow/form to be executed and adds the value to the current page/form. Example In online purchase, clicking 'Add to cart' will just add selected items to cart (executes the workflow and adds the value to the same form).
Link	On clicking the link, the target workflow/form will be invoked and executed.

Others

Data Table	Shows the table with data in it.
Attachment	To upload files. Appiyo Data Type (attachment) can be used for Attachment.
Image	To show images.

© Copyright 2014 Appiyo. All rights reserved.

Single Series Chart	Single Series Chart is a graphical representation that provides information about one item.
Multi Series Chart	Multi series chart is used to compare multiple items.
Title	To add a label to the group of menus following the title.
Tab Control	To add clickable tabs with group of menus inside each tab.

Icons

⊕	Add rows above/below.
⊖	Deletes the row.

Right click on form control

Delete	To delete the added form control.

Type Properties

Form

Click on the form label to modify the form properties on the right.

Name	Name of the form.
Description	Description of the form.
Title	Label of the form.
Grid Layout	To add row grids (known as spans) to the form table. For example, 12 span (maximum) = 1 row 6-6 span = 2 equal width rows
Label Alignment	Defines form label alignment.
Width	Defines the form width (in %).

Row

Click on the required row to modify its grid properties on the right.

Grid Layout	To add row grids (known as spans) to the selected row. For example, Inherit = Inherits the grid layout properties of the from table 12 span (maximum) = 1 row 6-6 span = 2 equal width rows

© Copyright 2014 Appiyo. All rights reserved.

Drag and drop the required form control from the left and click on it to modify the properties on the right.

Field

Label	Title of the form control.
Name	Assigned variable (defined inside workflow) name.
Hint	Hints on the information to be entered in the form control.
Value	Used to set a default value for the form control.
Label Alignment	Defines form control's label alignment. Inherit will follow the label alignment property of the form table.
Label Width (%)	Defines the label width (in %).
Disable	Restricts other users from editing the form control.
Show As Text	Shows the date and time saved in time zone format (10/30/2014 11:15 AM GMT+05:30) in the correct format (30-10-2014 11:15:00).
Link Label	Name of the **Link**.
Action	Action (**Invoke A Process**/**Invoke A Form**) to be taken on clicking the **Link**.
Target	Process/Form to be invoked on clicking the **Link**.
Service Input	Adds the form controls that has to get the input from the specified process on clicking the **Link**.
Operation (Upload/Download)	Operation to be performed on the **Attachment** file.
Process	Process that has to be executed for the file **Attachment**.
Multiple Upload	Allows to upload multiple files under **Attachment**.
Dynamic	To get the values for **Src** and **Text** dynamically i.e. during process execution. **Src** and **Text** should be mapped with respective variables in workflow.
Src	Source link of the **Image** to be shown.
Text	Text to be displayed under **Image**.
Chart Type	Chart model (Pie, Bar etc.) to be shown.
Category	Defines the values for the x-axis of the Multi series chart.
Series	Provides data (i.e. y-axis values) for the x-axis of the Multi series chart.

© Copyright 2014 Appiyo. All rights reserved.

Help Text	Gives information about the group of menus following the **Title**.

Validators

Checks whether all mandatory fields in the form are filled and shows the defined **Error Message** (if fields are empty/incorrect). Click **Edit** to add validators.

Required	Sets the field as required field. Example – Text box to enter name.
Max. no of characters	Sets maximum limit to enter the characters. Example – Text box to enter address.
Min. no of characters	Sets minimum limit to enter the characters. Example – Text box to enter new password.
Equality	Compares the value entered with the defined field value. Example – Text box to enter password for login.
Pattern	Compares the entered value with the defined patterns like **Email** – @example.com **URL** – http/https://www.example.com **Pattern** – User defined **Expression**
Maximum value	Sets maximum limit for the value to be entered. 3 = Values less than or equal to 3 can be entered.
Minimum value	Sets minimum limit for the value to be entered. 3 = Values greater than or equal to 3 can be entered.
Max. no of digits	Sets maximum limit to enter the number digit. 3 = Values can be in three digit or less.
Min. no of digits	Sets minimum limit to enter the number digit. 3 = Values can be in three digit or greater.

Validator Properties

Click on the required validators to define their properties on the right.

Error Message	Message to be shown on empty field or incorrect value.
Value	Maximum/minimum limit for the value to be entered.
Compare with	Field to compare with the entered value.

© Copyright 2014 Appiyo. All rights reserved.

Type	Defines the pattern type (**Email**, **URL** and **Pattern**) to be compared with the value.
Expression (Type Pattern)	User can define expression to compare with the entered values. '^[A-Z]' means value entered should start with uppercase alphabet between A-Z

Options (Select Box and Radio Button)

Click **Edit** to modify the options of Select Box/Radio Button.

Allow Multiple Selections	Allows to make multiple selections.
Variable for List of options	To get values from the collection variable (datatype) defined inside workflow.
Label	Name of the option to be included in the list.
Value	Set a value for the options in the list.
Selected	Selected options will be highlighted in the list.

Configuration (Autofill)

Click **Edit** to modify the properties of Autofill.

Invoke process on change	Invokes the **Target Process** on changes made to the **Autofill** field.
Target Process	Process to be invoked on changes.
Search Variable	Assigned variable to get the value searched by the user and send to the respective workflow to get the matching option list (**Options Variable**).
Options Variable	Variable assigned for the option list.
Accept multiple values	**Yes** = Allows to select multiple options from the field **No** = Can select only one option
Accept values only from dropdown list	True = Allows user to select options only from the predefined drop down list False = Allows user to enter their own values

Configuration (Data Table and Tab Control)

Click **Edit** to modify the properties of Data Table and Tab Control.

Define title (**Label**) and assign variable (**Name**) for each and every column/tab of Data Table and Tab Control.

Data Table

Disable dynamic addition/removal of rows restricts other users from adding rows to the **Data Table**.

© Copyright 2014 Appiyo. All rights reserved.

On submit send table data to backend service will send back the data with the changes made by the user to update the table.

Enable pagination adds page number link (like 1, 2) to the table. This can be used when large contents cannot be shown in single page.

Pagination

Process	Process that has to be executed to get the data to be shown in the table.
Total Pages	Specifies the total number of pages of the table in the page number link. Since this is dynamic it will be mapped with respective variables in the workflow to get its value during execution. For example if table has 10 total pages, page links from 1 to 10 will be shown.
Number of page links	Specifies the number of page links to be shown at a time. For example, specifying 5 (out of total pages 10) will show page links from 1 to 5 at a time.
Page Number	Specifies the current page number. User clicked page number has to be sent to the workflow to get the data of the requested page. So it has to be mapped with respective variable in the workflow.

Note: Total width of all the columns should not exceed 100.

Alert Box (Submit and Button)

On **Submit/Button** click opens an alert box with the defined **Message**.

Show alert on submit/click	True = Opens the alert box False = Do not open alert box
Header	Title of the alert box.
Message	Message to be shown in the alert box.
Label for button - Yes	Name of the first button (Yes, Ok).
Label for button - No	Name of the second button (No, Cancel).

Chart Properties

For more information on charts, refer Charts.

Caption	Title of the chart.
Sub Caption	Sub title of the chart.
Bg Color	HTML color code that defines the background color for the chart.
Format Number	Data will be shown in correct number format. Example: 180000 will be shown as 180,000

© Copyright 2014 Appiyo. All rights reserved.

Format Number Scale	Number scales can be used to represent lengthy data Example: 180000 will be shown as 180k where k = 1000
Number Prefix	Adds entered value as prefix to Y axis data $ can be added as prefix to indicate money
Number Suffix	Adds entered value as suffix to Y axis data C^o can be added as suffix to define temperature
Show Tool Tip	Shows information about the items in the chart when cursor is placed over it
Xaxisname	Label for the X axis
Yaxisname	Label for the Y axis

Events

Defines the action to be taken when an event (**On** a form **Load**, **On** a button **Click** or **On** a value **Change**) occurs. Click **Add Event** to define **Event Property**.

Action Service

Service calls the specified process (workflow) when an event occurs.

Target Service	Add the process to be called.
Service Input	Add the form controls that has to get the input from the specified process.
Service Output	Add the form controls in which the output value of the specified process has to be stored.

Action Enable/Disable

Makes the specified form control enabled/disabled when an event occurs.

Service Output	Add the form controls to be enabled/disabled.

Note:

Form controls can also get values from the defined variables by mapping them with respective variables.

A unique id assigned to the created form (numbers attached at last with the url) is used in User Task.

Example: https://www.appiyo.com/#forms/**538ee14f756d27efb06fba89**.

© Copyright 2014 Appiyo. All rights reserved.

Configuration

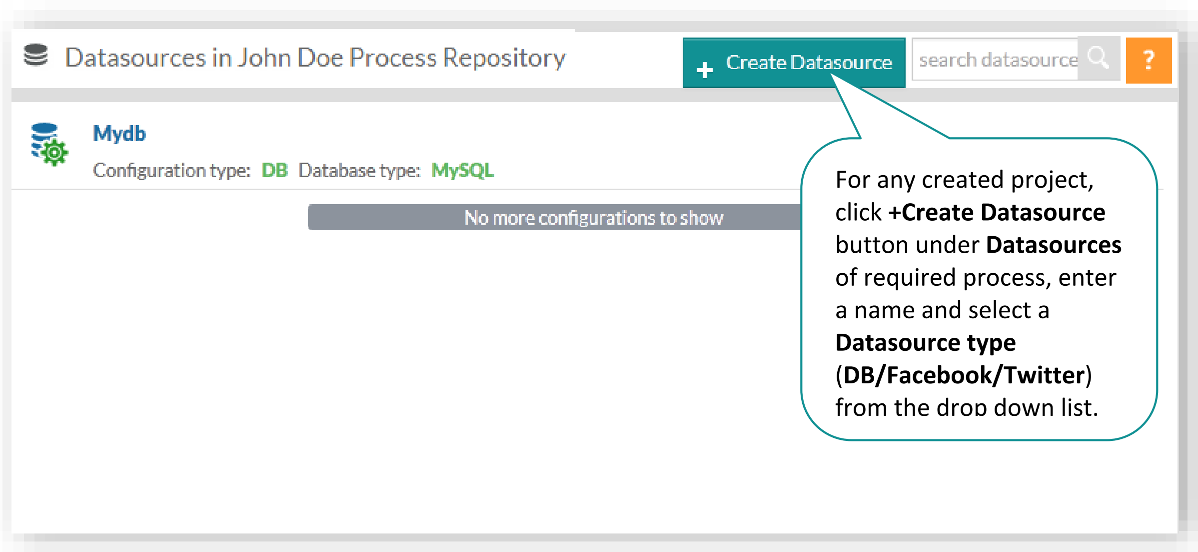
Datasources

- **Data sources** refer to any device or system which can provide data or receive data required for a workflow.
  - Access to such data sources may be provided through published REST APIs or Web Services.
  - Alternately, standard database access services such as JDBC/ODBC may be used.
  - Custom services may be necessitated in legacy systems.
  - REST APIs are increasing in adoption as a standard which will enable wider access to a variety of data sources.
  - Some examples of data sources include social media, weather forecast web service and salesforce system.

Create Datasources

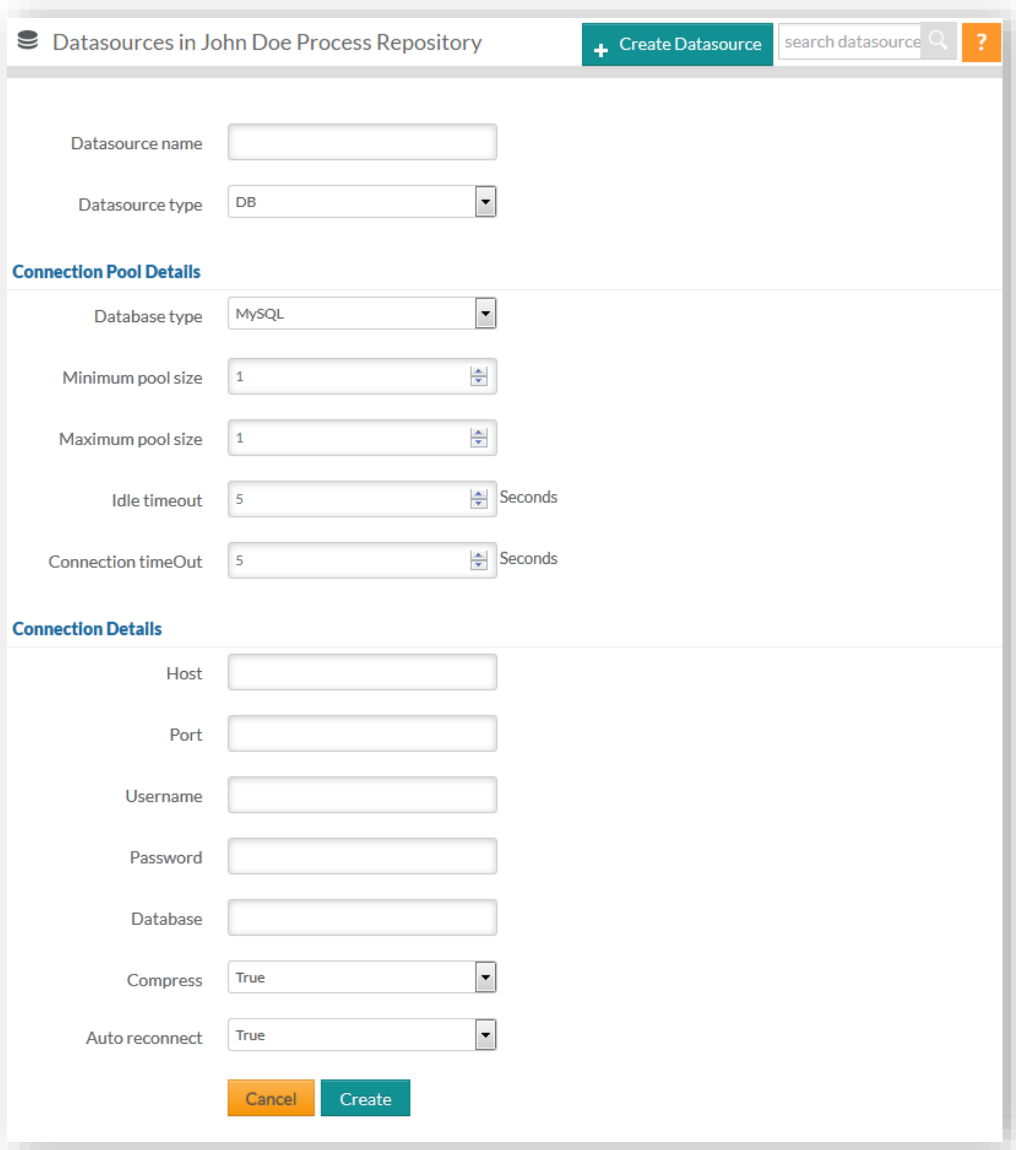
Create a new datasource in your process. Datasource includes defined database connectivities, Facebook and Twitter connectivity for your process.

Navigation: Required process repository → Configuration tab → Datasources



© Copyright 2014 Appiyo. All rights reserved.

DB



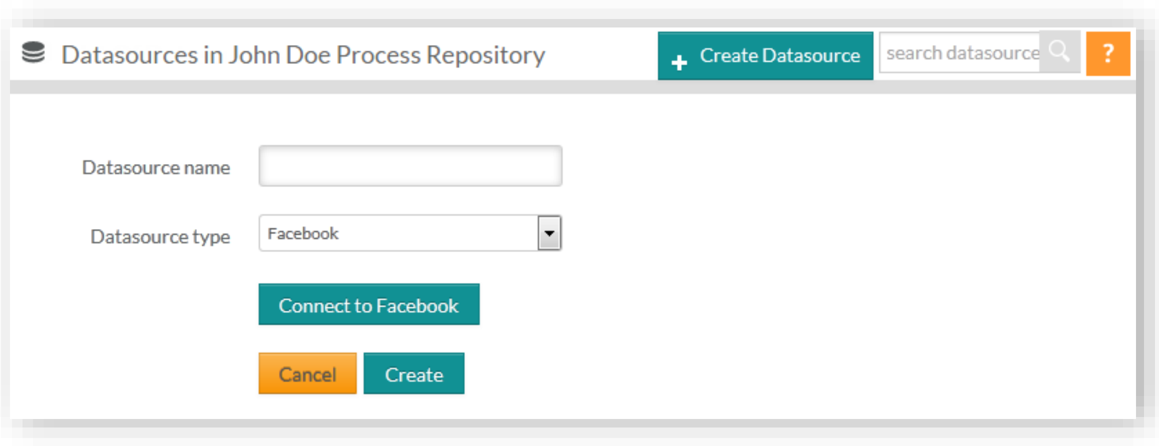
Select a **Database type** (**MySQL/PgSQL/SQLServer**) from the drop down list and enter appropriate details under **Minimum pool size**, **Maximum pool size**, **Idle timeout** and **Connection timeOut**.

Include host and port url under **Host** and **Port** respectively and enter username and password with the database name.

Select an option (**True\False**) from the drop down list for **Compress** and **Auto reconnect** and click **Create.**

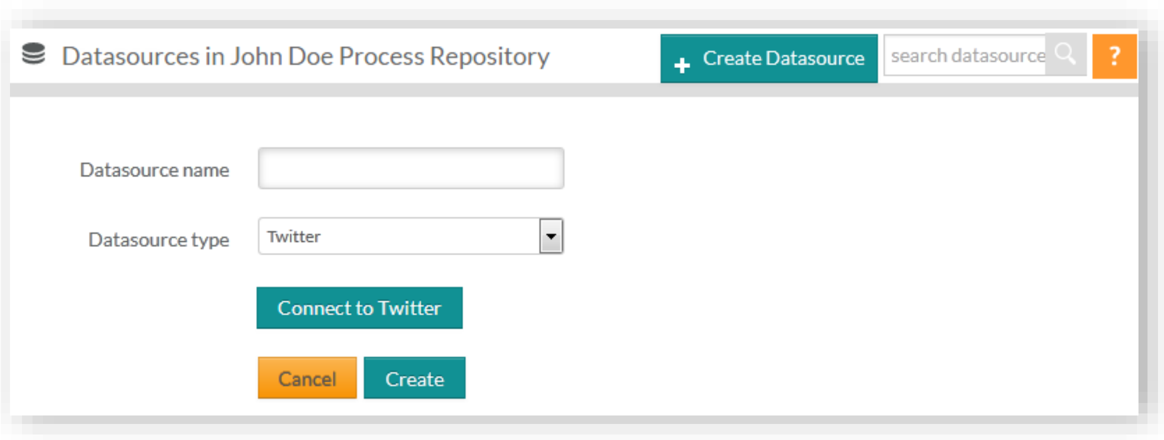
© Copyright 2014 Appiyo. All rights reserved.

Facebook



Click **Connect to Facebook**, login with your id and password and click **Create**.

Twitter



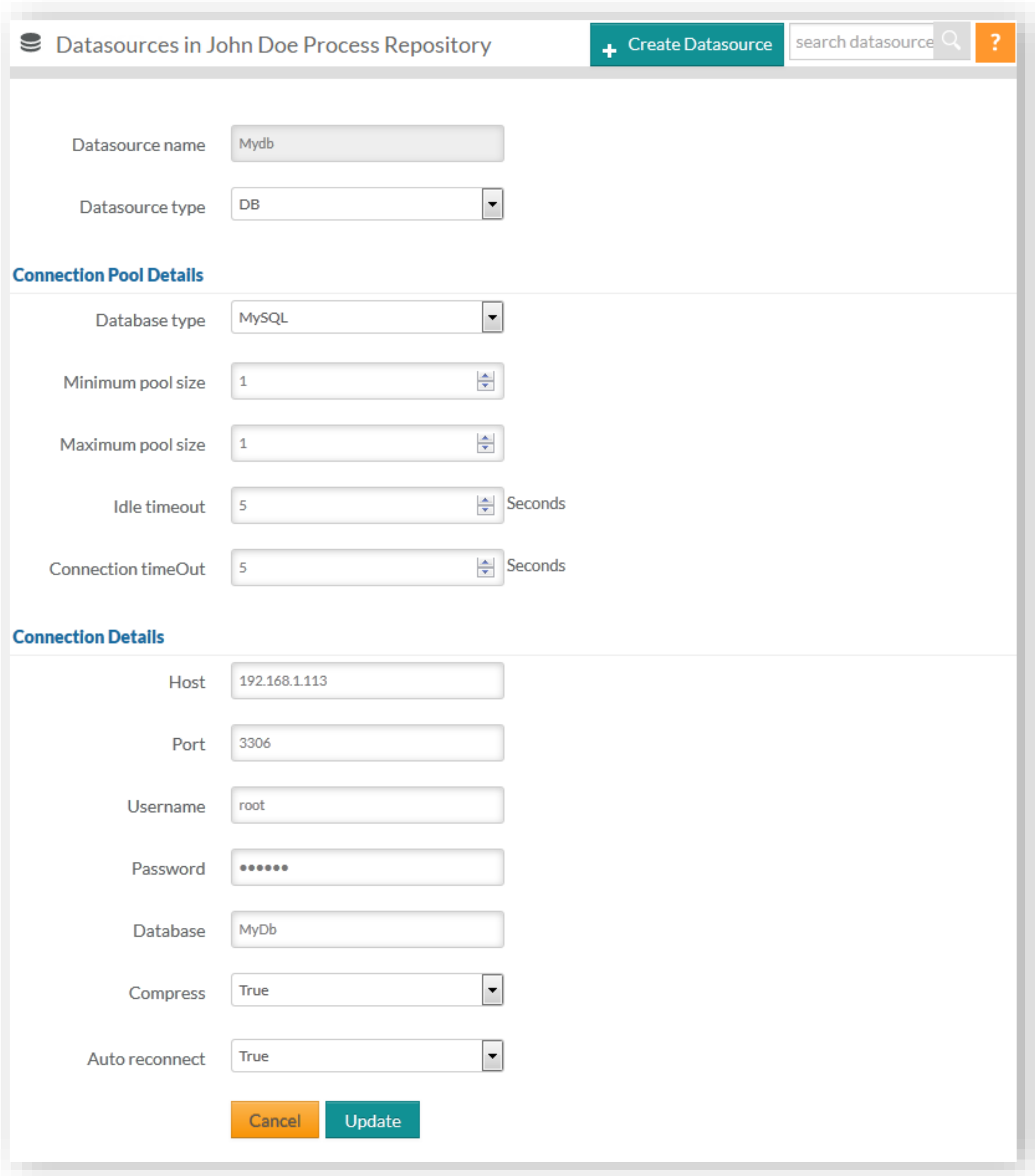
Click **Connect to Twitter**, login with your id and password and click **Create**.

© Copyright 2014 Appiyo. All rights reserved.

Update Datasources

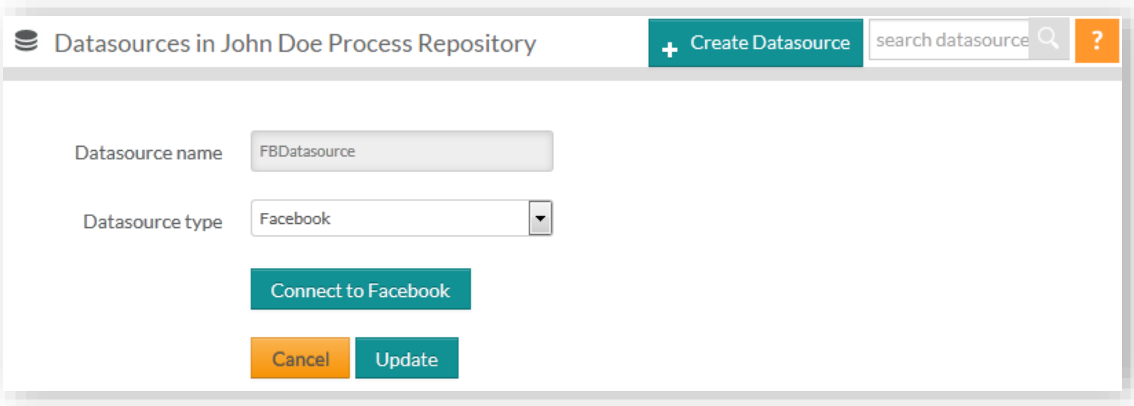
To **modify your datasource**, click the datasource to be modified, make necessary changes and click **Update**.

DB

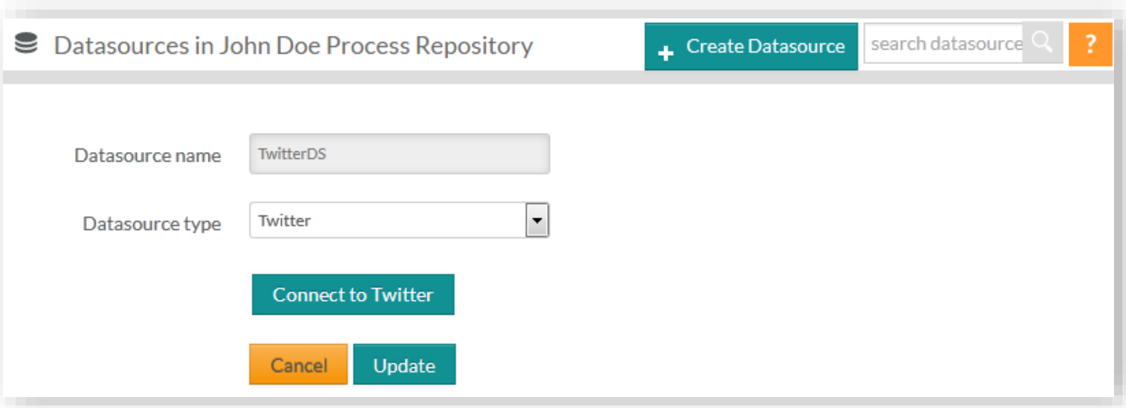


© Copyright 2014 Appiyo. All rights reserved.

Facebook



Twitter



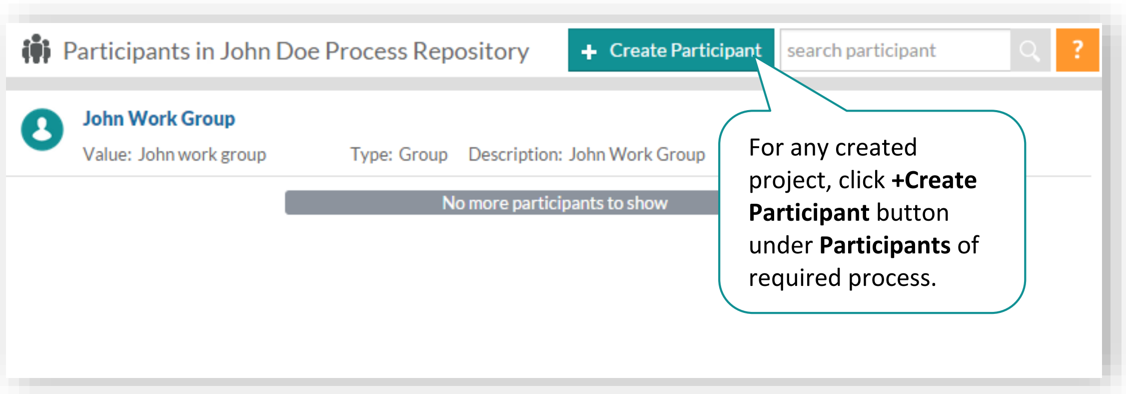
© Copyright 2014 Appiyo. All rights reserved.

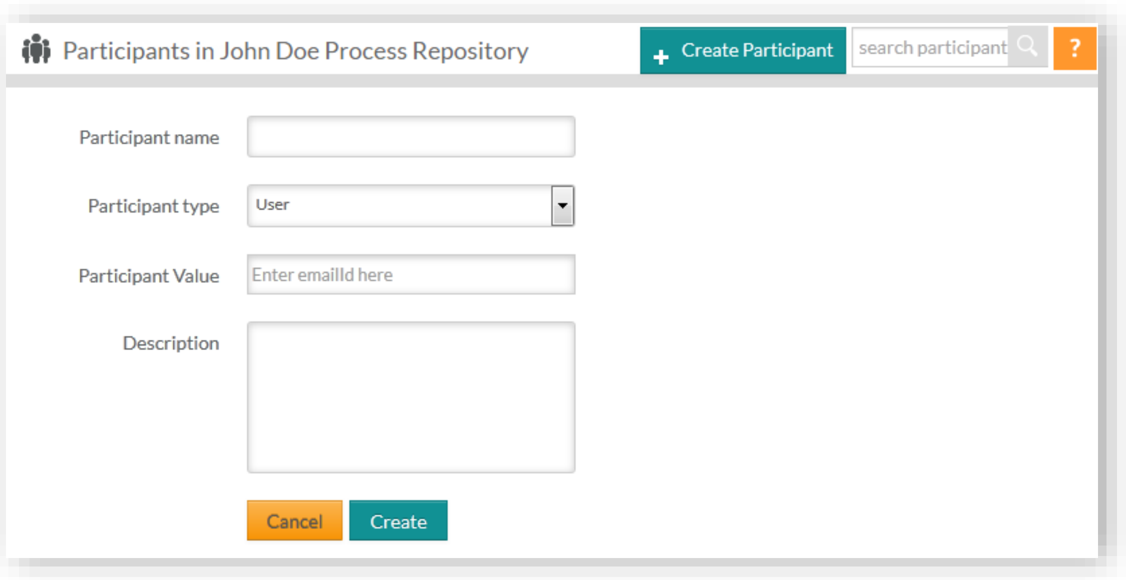
Participants

- Manual task in a workflow (process) is executed by a person. Since persons may change in different implementations, the concept of **participants** has been introduced.
  - Participants are placeholders for actual task performers.
  - Here you map an individual or a group to a participant. Examples of participants are 'Recruiter', 'Authorizer', 'Interviewer' and so on.
  - Participant is a useful concept which eliminates the need to make changes to workflows to specify workers.
  - Before deploying a process, you need to map your specific members to the participants defined in that workflow.

Participants include defined potential owners (users and groups) that can be used in your process for assigning tasks using User Task.

Navigation: Required process repository → Configuration tab → Participants

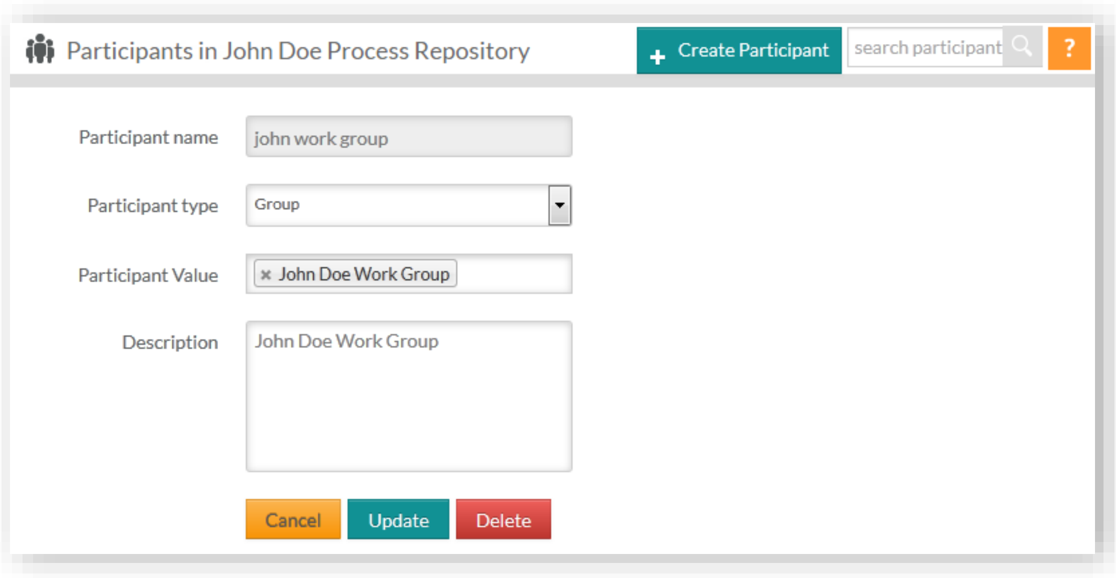




Enter a name and select a **Participant type** (**User/Group**) from the drop down list.

Enter **Participant Value** (user's email id/group id) with description and click **Create**.

© Copyright 2014 Appiyo. All rights reserved.



To **modify participants**, click the required participant, make necessary changes and click **Update**.

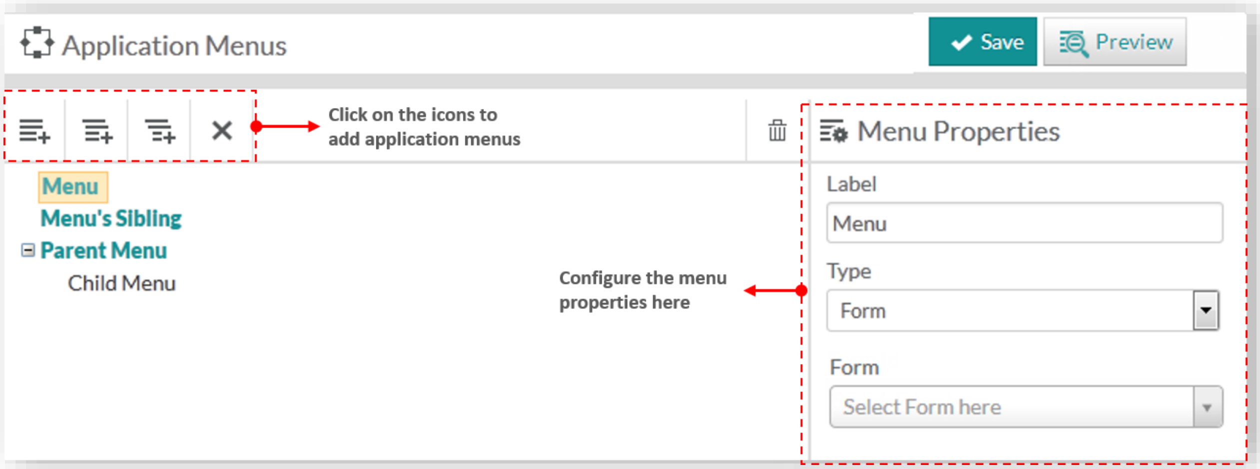
To **delete created participants**, click the required participant and click **Delete**.

© Copyright 2014 Appiyo. All rights reserved.

App Menus

Design User Interface (UI) for your application (with created forms) through which the users can interact and execute your application.

Navigation: Required process repository → App Menus tab



Add Menu adds the specified **Form** under the given **Label** tab.

Add Sibling adds the specified **Form** under the given **Label** tab. Sibling can be added only under **Menu**.

Add Child adds the specified child menu as a sub menu under the drop down list of the parent menu (to which the child menu is added).

Deletes required menus.

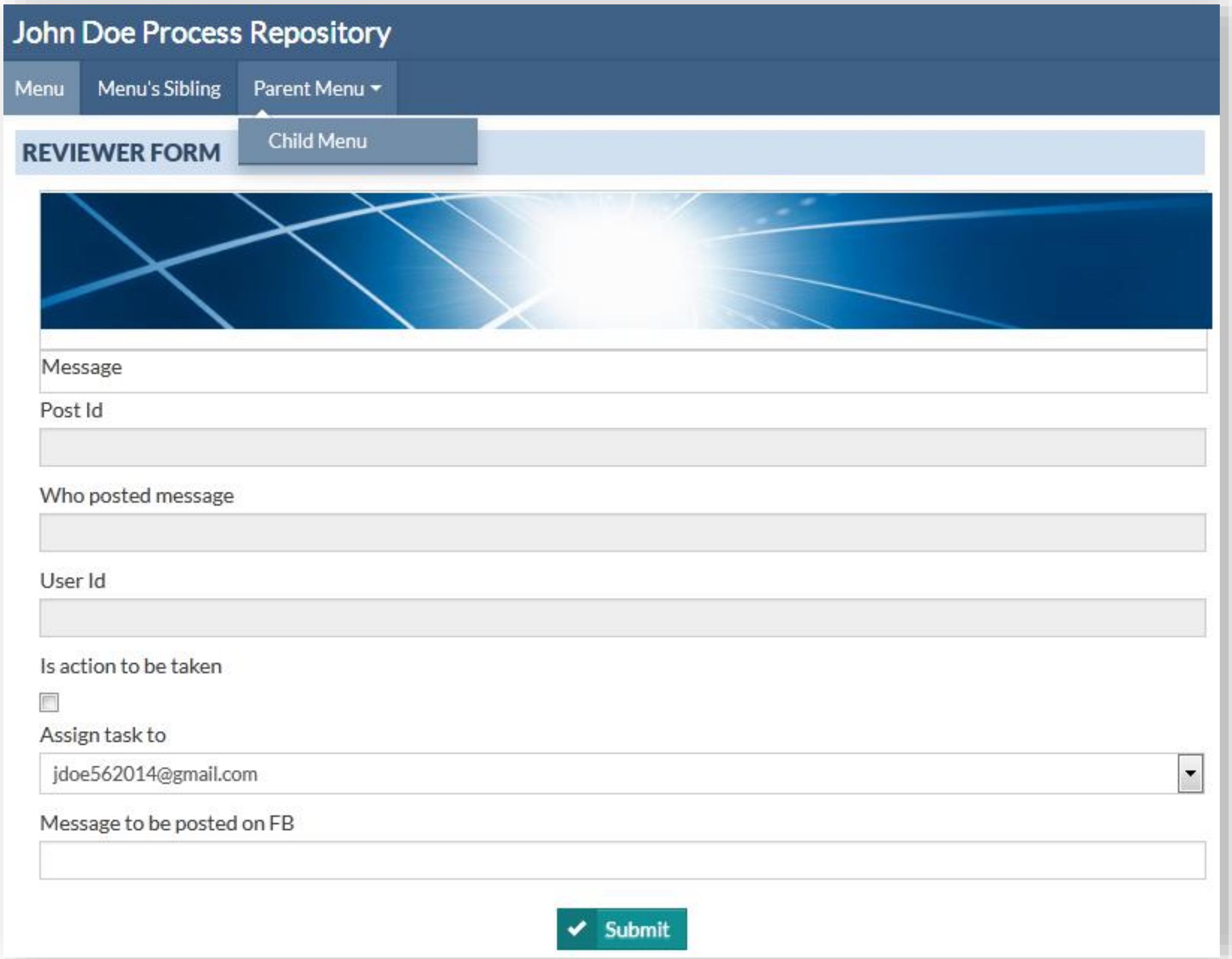
Deletes all the added menus.

© Copyright 2014 Appiyo. All rights reserved.

Execute

Execute your application with the designed application UI.

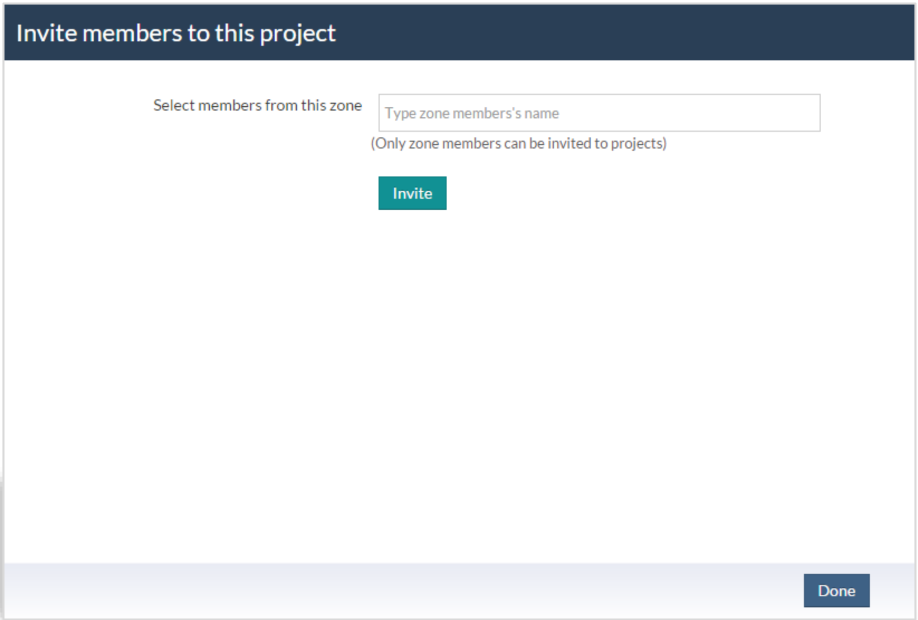
Navigation: Required process repository → Execute tab



© Copyright 2014 Appiyo. All rights reserved.

Invite to process repository

You can use this feature to invite your zone users to join your process repository.



To **invite** zone users **to join your process repository**, click **Invite** under **Description** tab of required process repository, enter the user's name and click **Invite.**

Subscribe to process repository

You can use this feature to accept the invitation and subscribe to the process repository.

To **subscribe to a process repository**, click **Accept** of required process repository under **Home** (Notification).

Reject process repository invite

Reject a process repository invite allows you to reject the invite from a process repository.

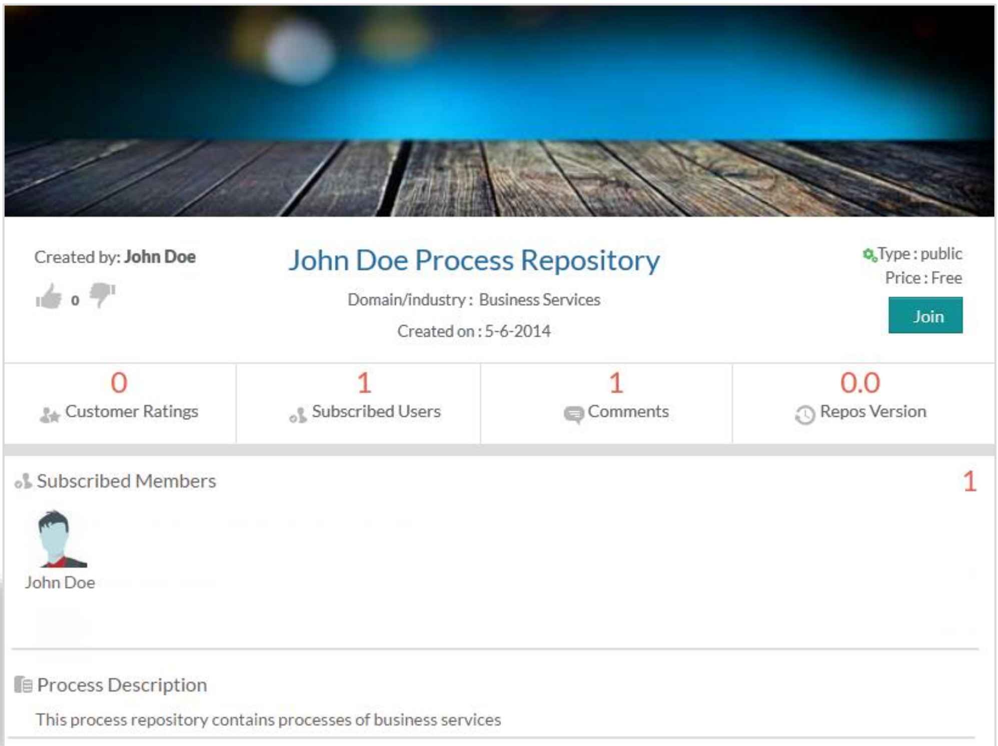
To **reject a process repository invite**, click **Reject** of the required process repository under **Home** (Notification).

© Copyright 2014 Appiyo. All rights reserved.

Join process repository

Join process repository allows you to send process repository subscription request to its owner.

Navigation: Process → Explore → Required Process Repository → Join



To **send subscription request to process repository owner**, click **Join** of required process repository.

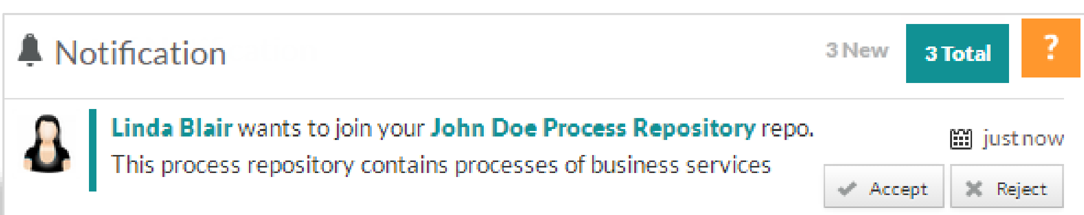
Approve process repository subscription request

You can approve the request of other users to join your process repository.

To **approve process repository subscription request**, click **Accept** of required process repository under **Home** (Notification).

Reject process repository subscription request

You can reject the subscription request of other users to join your process repository.



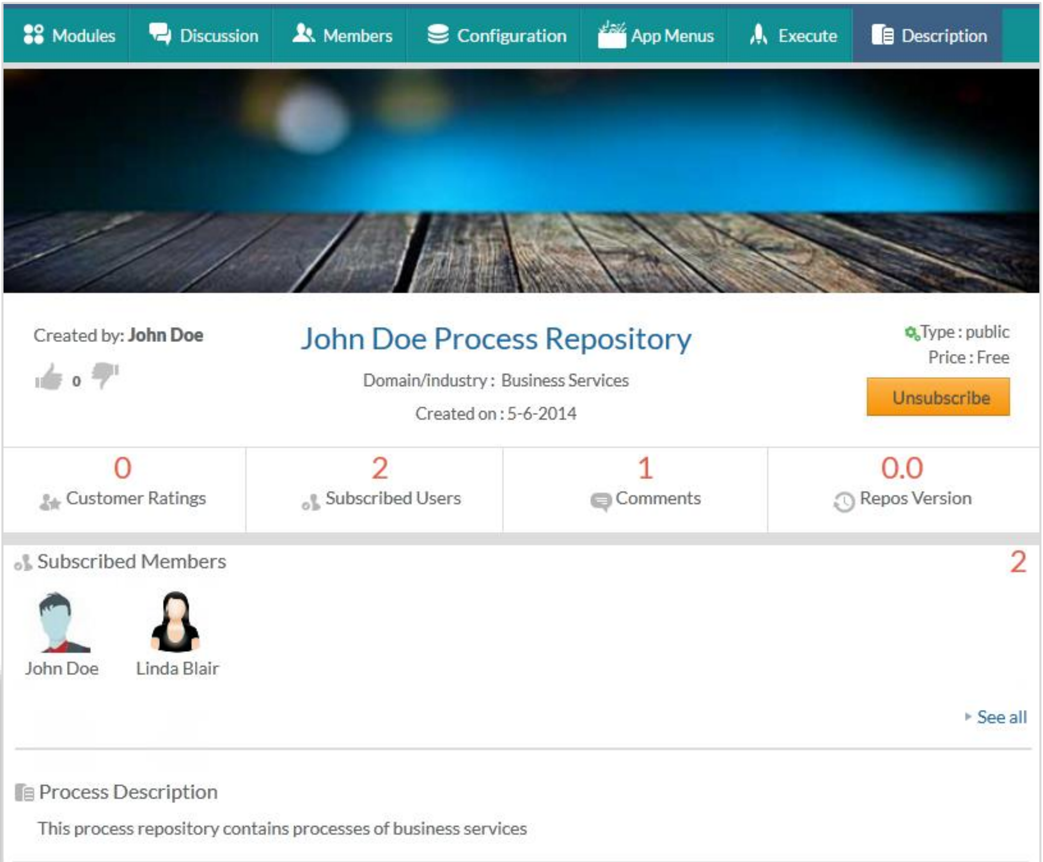
To **reject process repository subscription request**, click **Reject** of required process repository under **Home** (Notification).

© Copyright 2014 Appiyo. All rights reserved.

Unsubscribe process repository

Unsubscribe process repository allows you to leave the process repository.

Navigation: Required Process Repository → Description tab → Unsubscribe



To **unsubscribe a process repository**, click **Unsubscribe** under **Description** tab of required process repository.

Note: To join a process repo or to invite to a process repo in a zone, the user must belong to that zone i.e. the user must be zone user.

Up and Down Vote

Up Vote allows you to add your like to process repository.

To **add your like to process repository** click the **icon** from the required process repository.

Down Vote allows you to add your dislike to process repository.

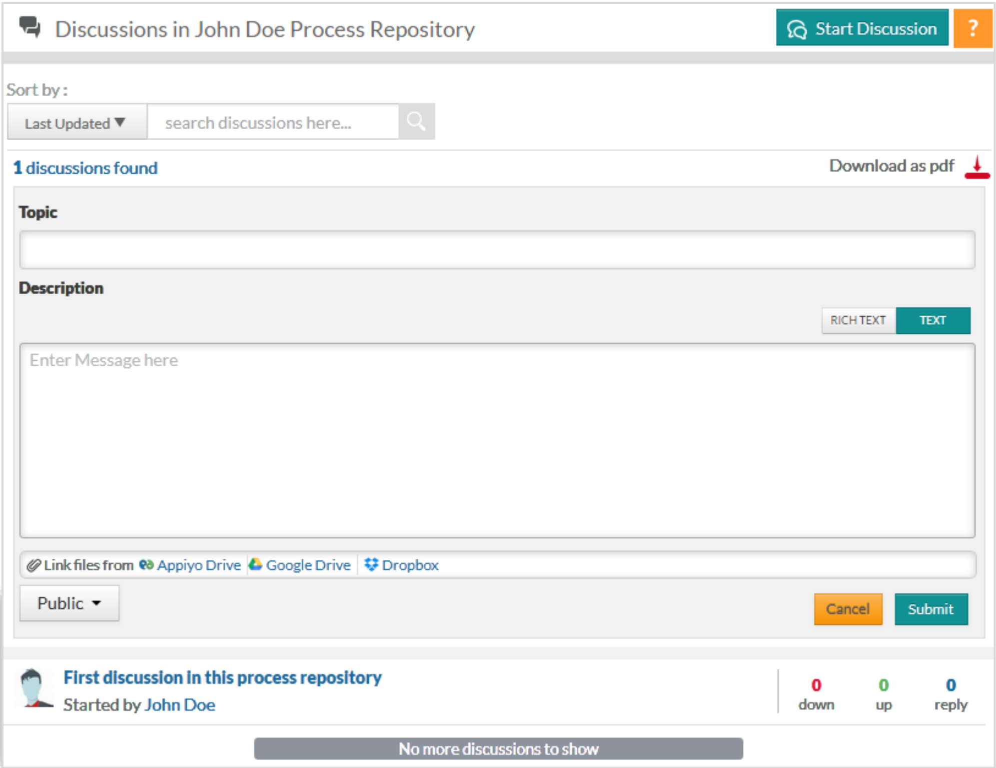
To **add your dislike to process repository** click the **icon** from the required process repository.

© Copyright 2014 Appiyo. All rights reserved.

Discussion

- If you want to collaboratively build your process apps, you will have to co-opt others and then '**Start Discussion**' to exchange thoughts.
  - Did it occur to you that you can quickly sketch a workflow by hand, photo shoot it on your mobile, upload into one of Appiyo drive, google drive or dropbox and link it for others to share their thoughts? Later the process owner can refer to it and formalize it in Appiyo bpm.
  - Or, you may do a video selfie, expressing your inputs for the process design and share it as a link.
- Try it. And share your experience.

Navigation: Required Process Repository → Discussion tab → Start Discussion



To **start a discussion** for any process repository, click **Start Discussion** under **Discussion** tab of the required process repository and enter necessary details under **Topic** and **Description**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

Set the type as **Public** or **Private.** You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit**.

To **notify** about a discussion to **all** process repository **members** (under discussion reply), click **Notify all**.

To **notify required members** (under Private discussions), include the members list in the textbox near **Private** button.

To **filter discussions**, search for discussions under **search discussions here**. You can also search for discussions by selecting either of the two options **Last Updated** or **Last Created**.

To **download** all the **discussions as a pdf** file, click **Download as pdf**.

© Copyright 2014 Appiyo. All rights reserved.

To **edit discussion, delete discussion, follow/unfollow discussion** and **up/down vote**, click on the required discussion.

To **modify your discussion**, click **Edit** from the discussion (to be modified), make necessary changes and click **Update**.

To **delete your discussion**, click **Delete** from the discussion (to be deleted) and click **yes**.

To **get notifications about discussions,** click **Follow** of the required discussion.

To **stop getting notifications about discussions,** click **Unfollow** of the required discussion.

To **add your like to discussion** click the **icon** from the required discussion.

To **add your dislike to discussion** click the **icon** from the required discussion.

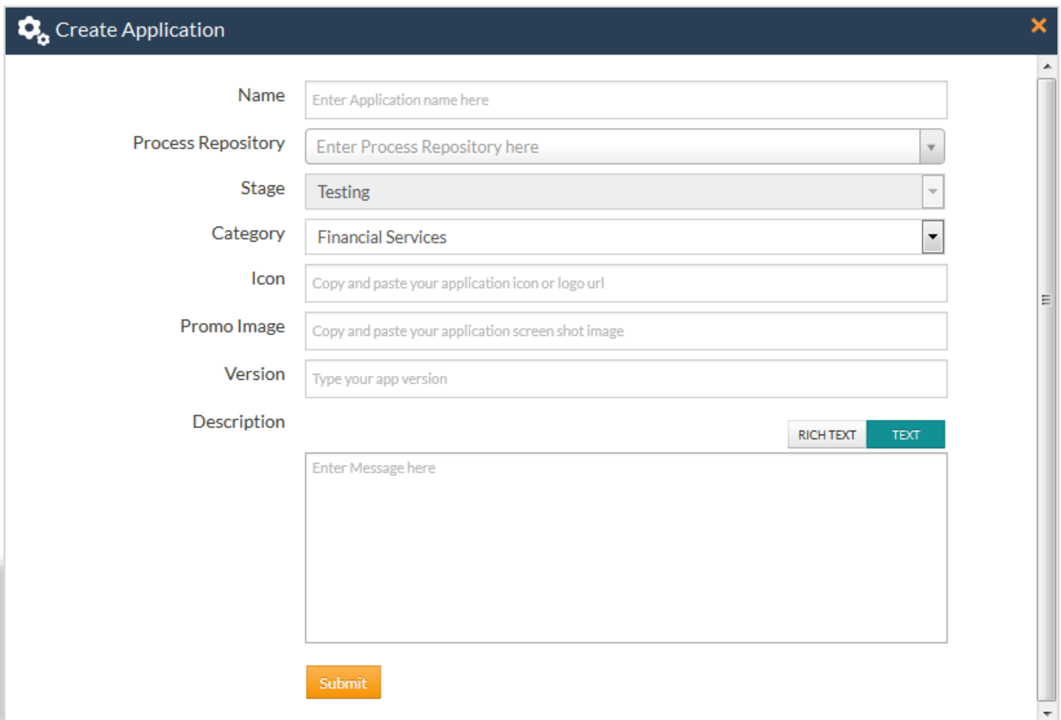
© Copyright 2014 Appiyo. All rights reserved.

Applications

- **Create Apps** readies an app for use. An app's usage may be for testing, for customization or for implementation. Create Apps is a useful concept to create clones of apps for different implementations. Typically, the following happen.
  - All the technical components (aka binaries) which make up an app are assigned to the new name of the app.
  - The zone administrator sets up datasources and participants for the app. This prepares the app for use in a particular context (also refer to Applications).

Create application

Navigation: Applications tab → Create an Application



To **create an application**, click **Create an Application** under **Applications.**

Enter a **Name** for the application, enter the **Process Repository** name from which the application has to be built and select the **Category** from the drop down list.

Paste the appropriate url under **Icon** (application's logo) and **Promo Image** (application's screen shot). Enter application's version under **Version** and descriptions under **Description.**

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

RICH TEXT – Includes formatting and hyperlink attachment options.

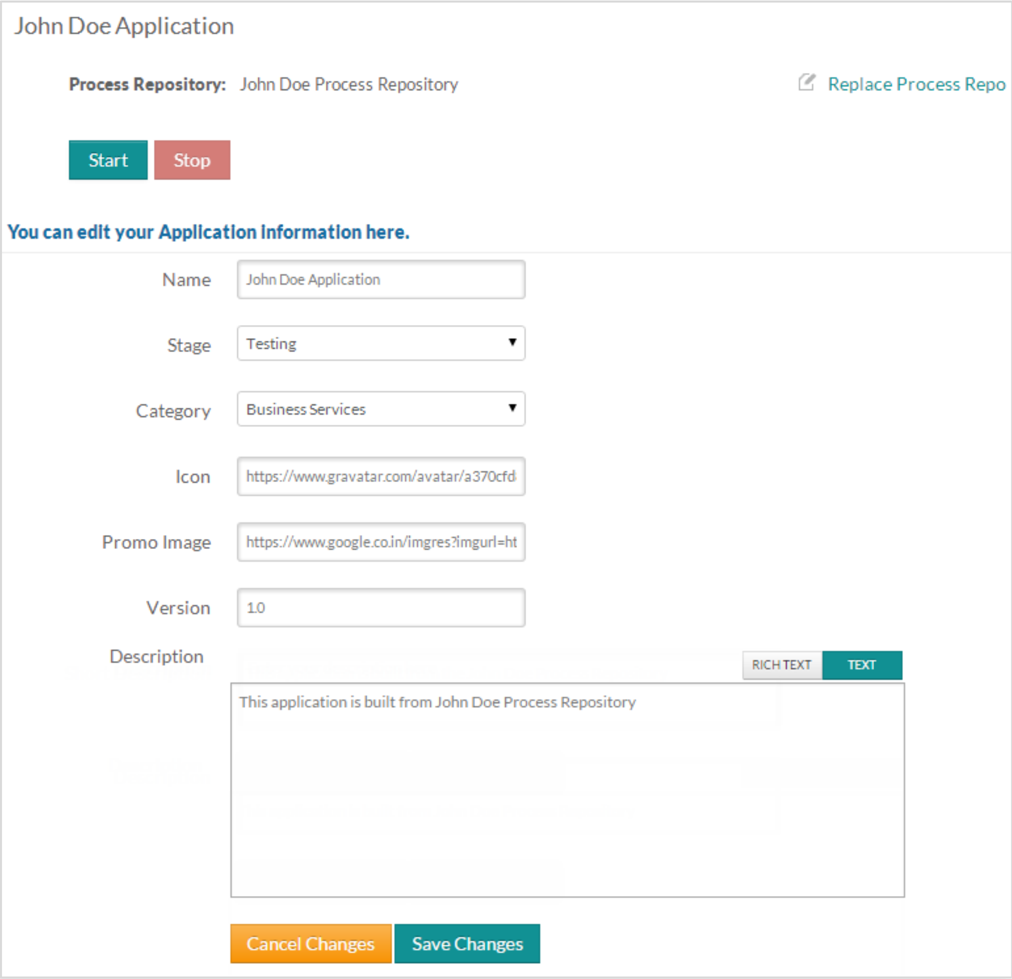
TEXT – Ignores formatting and hyperlink attachment options.

Click **Submit**, the application will be created and listed under Applications.

© Copyright 2014 Appiyo. All rights reserved.

Edit App allows you to modify your application details.

Navigation: Required Application → Edit App (under application cover photo)



To **modify your application details**, click **Edit App** of required application, make necessary changes and click **Save Changes.**

Start allows you to execute your application to test it. **Stop** allows you to stop the execution.

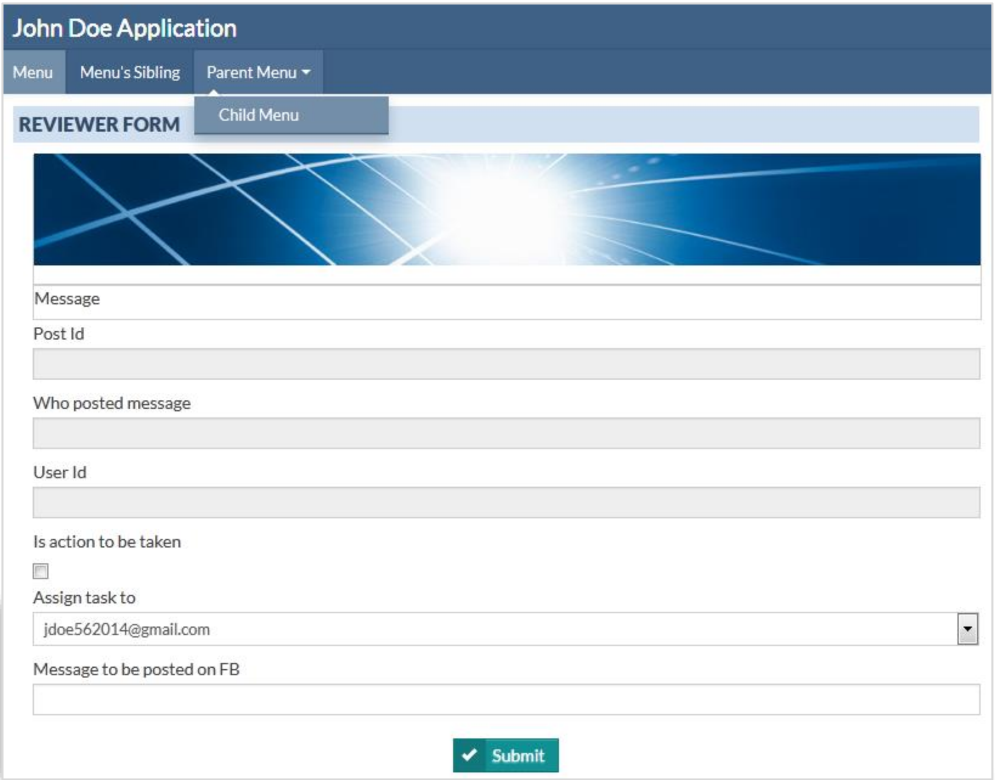
Replace Process Repo will update the application with the specified process repository.

© Copyright 2014 Appiyo. All rights reserved.

Execute App

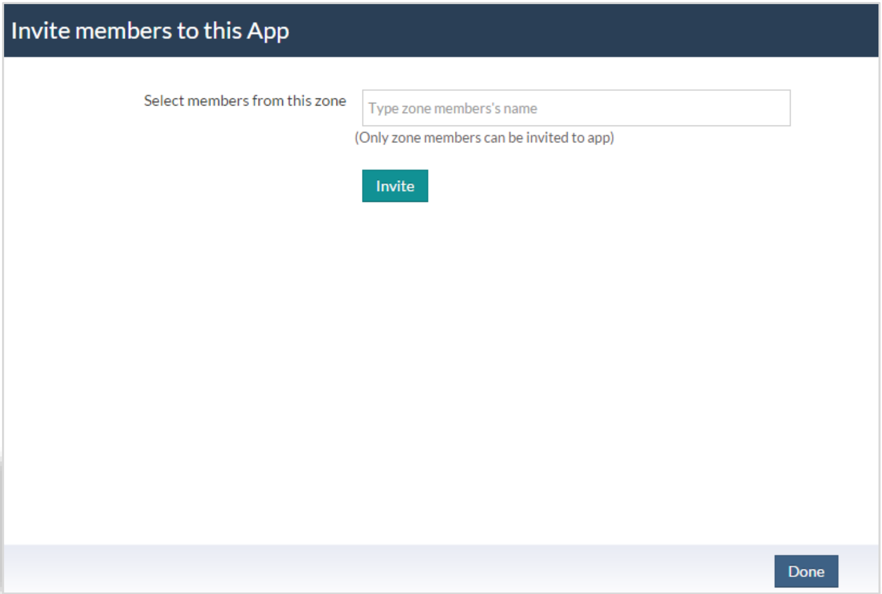
After clicking **Start** under **Edit App**, execute your application under **Execute** tab.

Navigation: Required Application → Start (under Edit App) → Execute tab



Invite to Application

You can use this feature to invite your zone users to join your application.



To **invite** zone users **to join your application**, click **Invite** under **Description** tab of required application, enter the user's name and click **Invite.**

© Copyright 2014 Appiyo. All rights reserved.

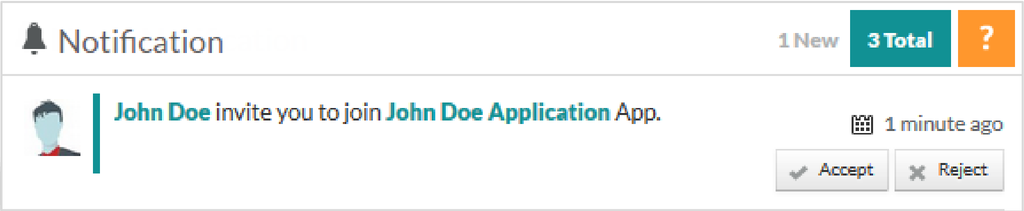
Accept application invite

You can use this feature to accept the invitation and subscribe to the application.

To **subscribe to an application**, click **Accept** of required application under **Home** (Notification).

Reject application invite

Reject an application invite allows you to reject the invite from an application.

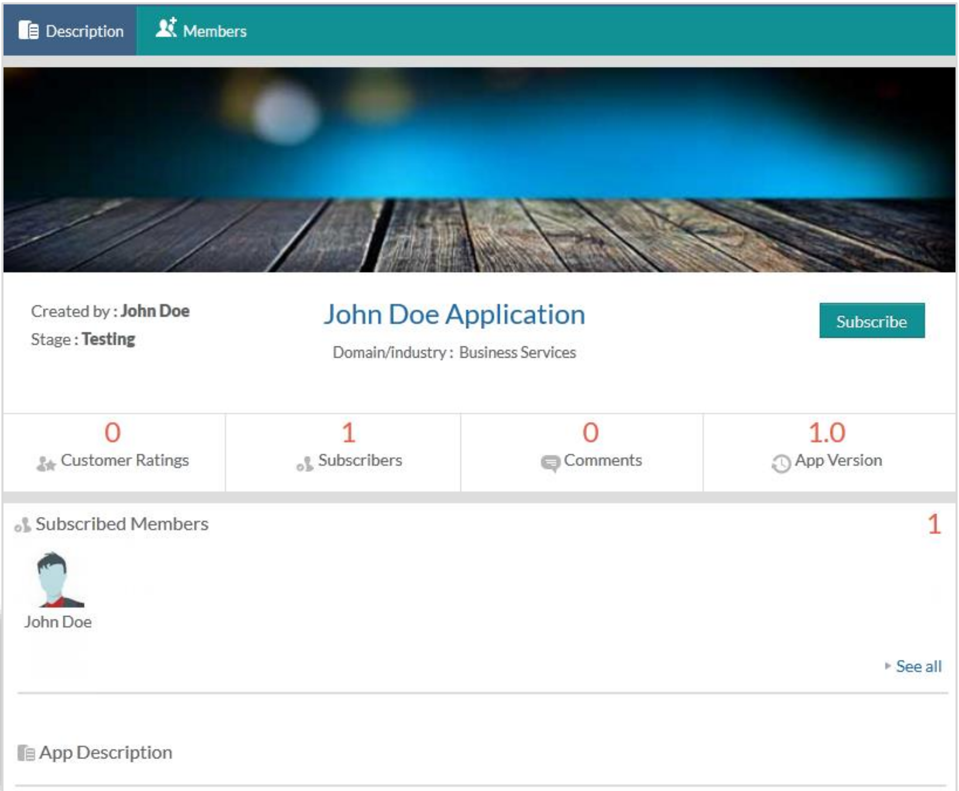


To **reject an application invite**, click **Reject** of the required application under **Home** (Notification).

Subscribe to applications

Subscribe to applications allows you to send subscription request to the application's owner.

Navigation: Applications → Explore → Required Application → Subscribe



To **subscribe to applications**, click **Subscribe** of required application under **Description** tab.

© Copyright 2014 Appiyo. All rights reserved.

Approve application subscription request

You can approve the request of other users to subscribe to your application.

To **approve application subscription request**, click **Accept** of required application under **Home** (Notification).

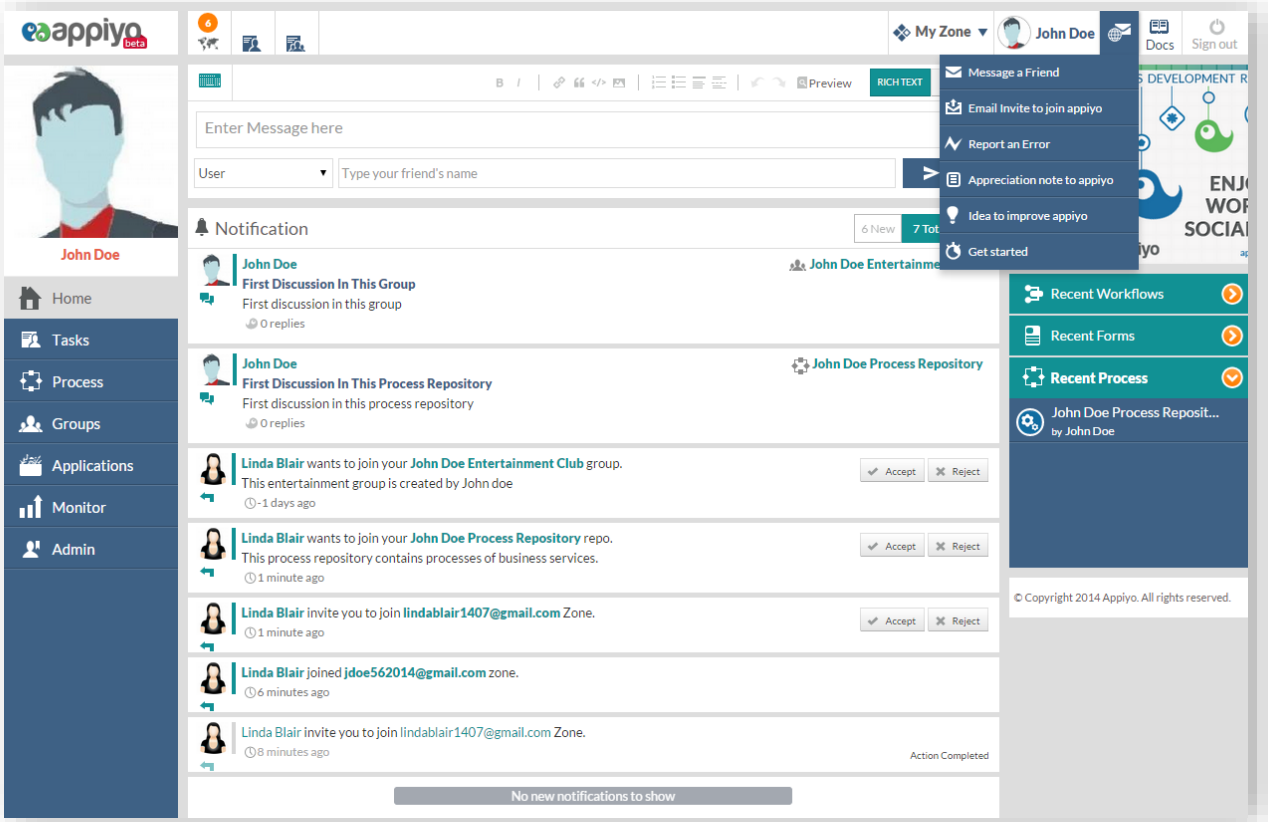
Reject application subscription request

You can reject the request of other users to subscribe to your application.

To **reject application subscription request**, click **Reject** of required application under **Home** (Notification).

© Copyright 2014 Appiyo. All rights reserved.

Appiyo Home Page



Home

- Notifications are shown here. These will also be available on your mobile.
  - Notifications may be information and some action items like approval of member request to join, for example.
- **Zones** are a powerful feature in Appiyo. They are clearly demarcated virtual organizations on cloud. They provide the first level of privacy and security. A zone is a virtual boundary under the control of the zone administrator who controls and secures access to the zone and resources within it.
  - Every Appiyo member gets an exclusive **community zone (My Zone)**. It is lifetime free.
  - Community zones will receive advertisements.
  - **Commercial zones** will be paid for, scalable with need and will be ad free.
- **Top menu bar** is an 'always-on' convenience provided to quickly attend to notifications, tasks and move across zones.

Post your message as notification on required **User** or **Group**.

Enter your **Message** and select **User** or **Group** from the drop down list.

Type your friend's/group name and click **Post**.

© Copyright 2014 Appiyo. All rights reserved.

The **icon** located on the top right corner of the screen includes the following features:

- Message a Friend
- Email Invite to Join appiyo
- Report an Error
- Appreciation note to appiyo
- Idea to improve appiyo
- Get started

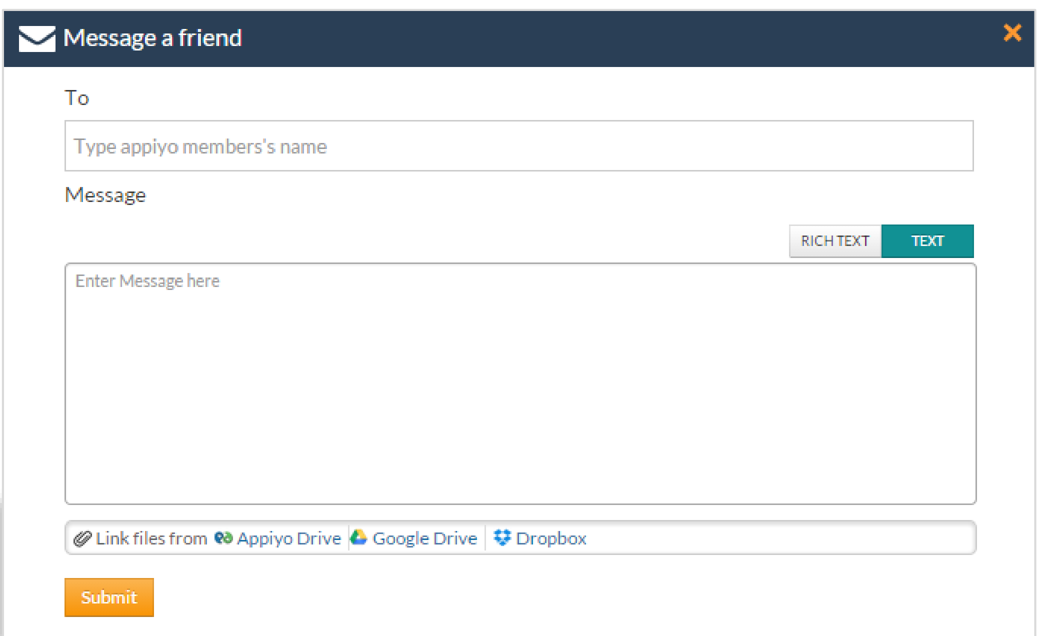
Message a Friend

You can send messages to your Appiyo friends. You can also send details about your social groups and process to promote them.

To message a friend, enter the name and the message body.

For message body, you can choose either **RICH TEXT** or **TEXT** option located at right side.

You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit.**

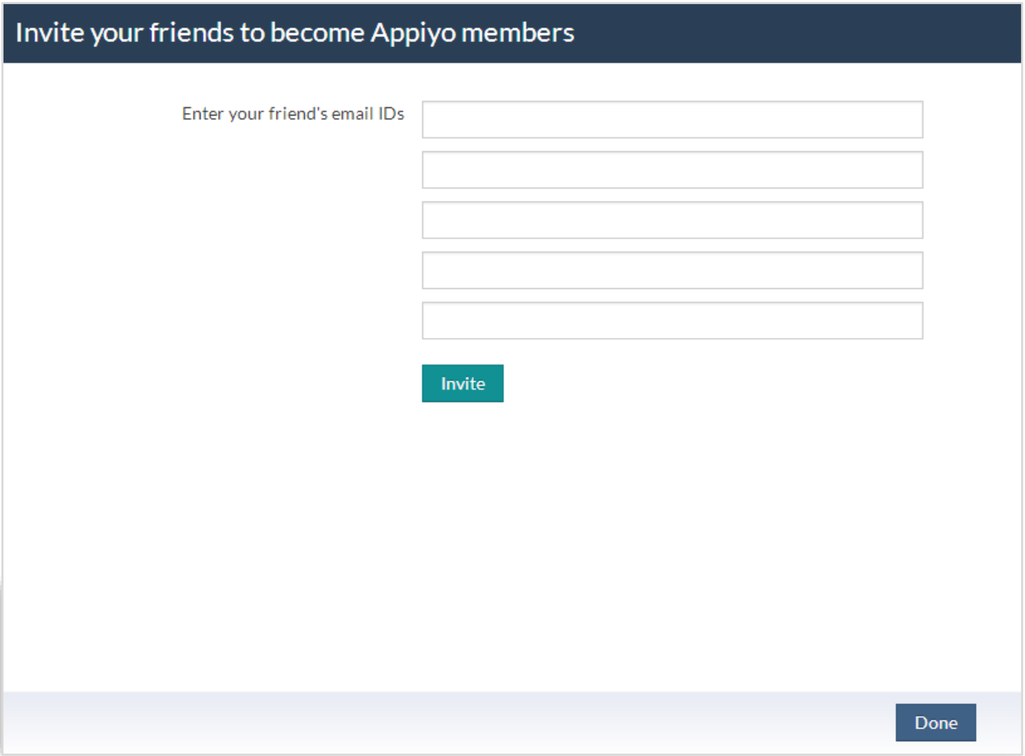


© Copyright 2014 Appiyo. All rights reserved.

Email Invite to Join appiyo

Send email invitations to your friends to join Appiyo. This helps you socialize more and increase the awareness about your business in the society.

To invite your friends to join Appiyo, enter the email address of your friend and click **Invite.**

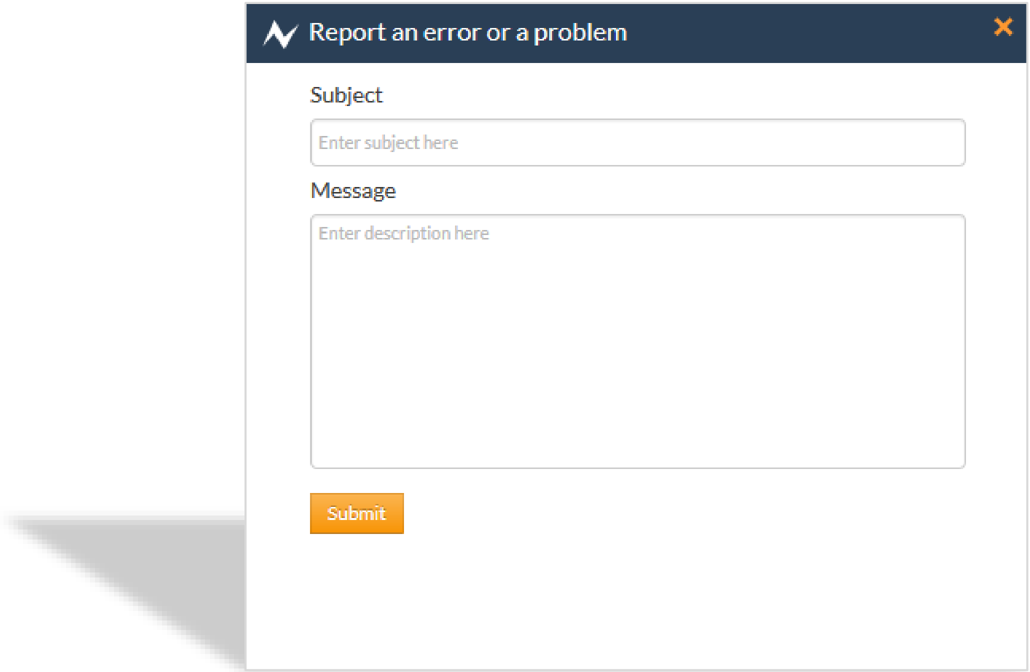


© Copyright 2014 Appiyo. All rights reserved.

Report an Error

Using this feature you can report any problems or errors to our developers while using the platform. This helps us enhance our platform and make the GUIs more user-friendly.

To report an error, enter appropriate details under fields **Subject**, **Message** and click **Submit.**

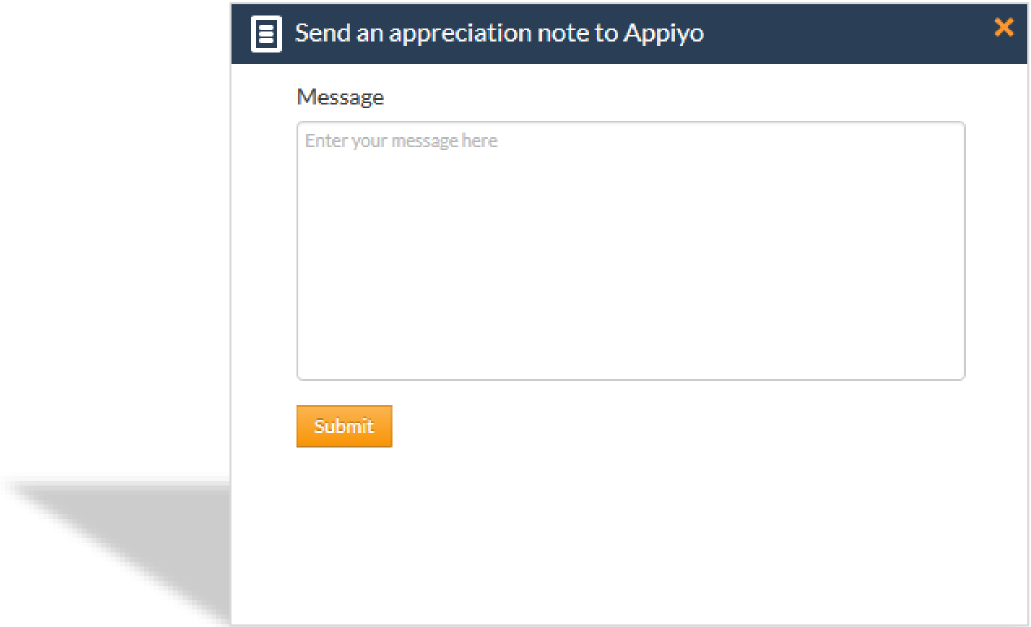


© Copyright 2014 Appiyo. All rights reserved.

Appreciation note to appiyo

Tell our developers about the Appiyo feature that you like by sending your appreciation notes.

To send an appreciation note, enter your appreciation message and click **Submit.**

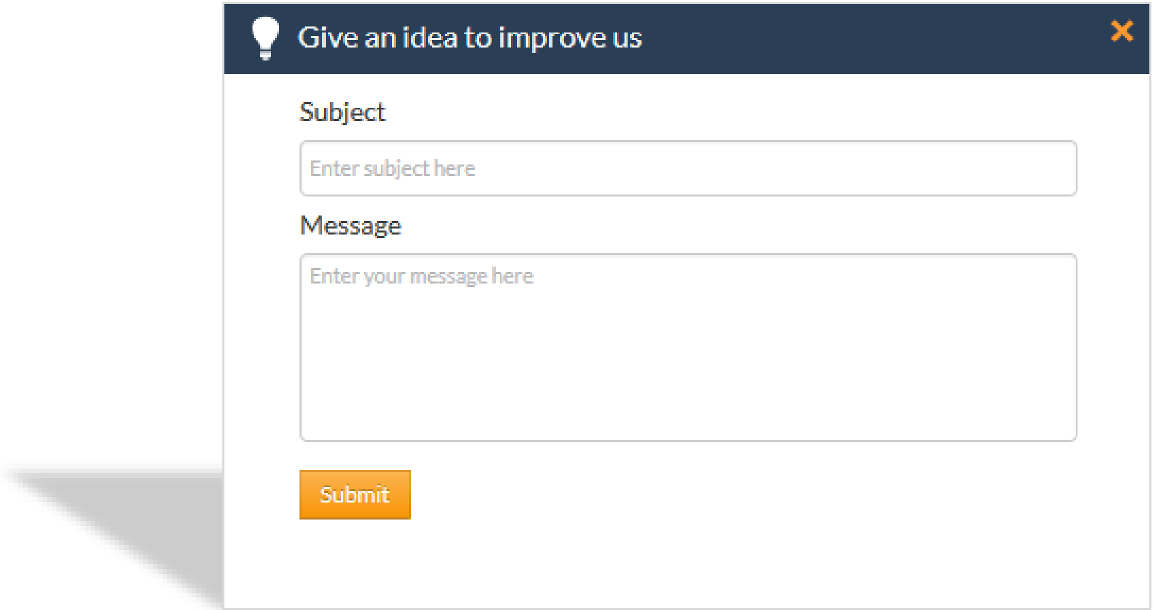


© Copyright 2014 Appiyo. All rights reserved.

Idea to improve appiyo

Send your innovative ideas to help Appiyo developers enhance the application.

To give an idea to improve, enter appropriate details under fields **Subject**, **Message** and click **Submit**.



© Copyright 2014 Appiyo. All rights reserved.

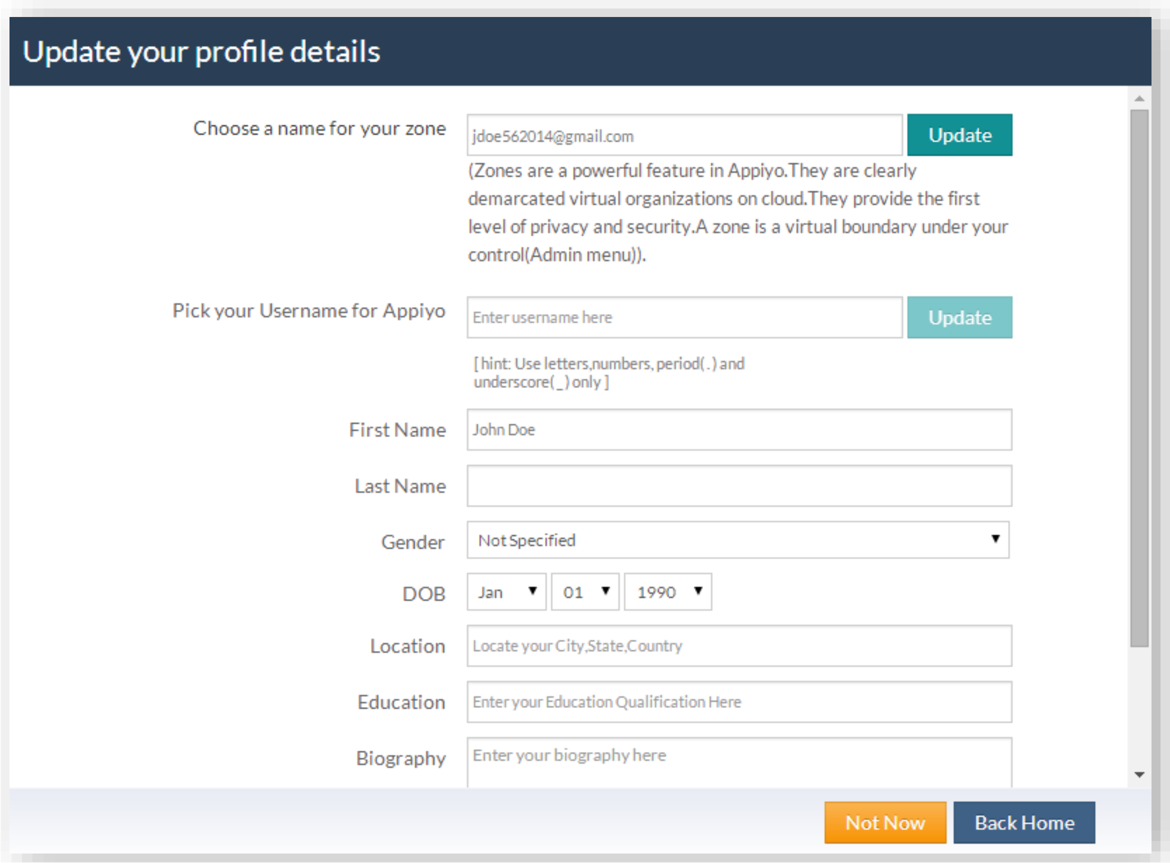
Get started

You can update your zone and user profile here (refer to **Setup Profile**).

To **update** your **name for your zone**, **Choose a name for your zone** and click **Update**.

To **update** your **name for Appiyo**, **Pick your Username for Appiyo** and click **Update**.

To **update** your **user profile**, enter appropriate details and click **Save Changes**.



© Copyright 2014 Appiyo. All rights reserved.

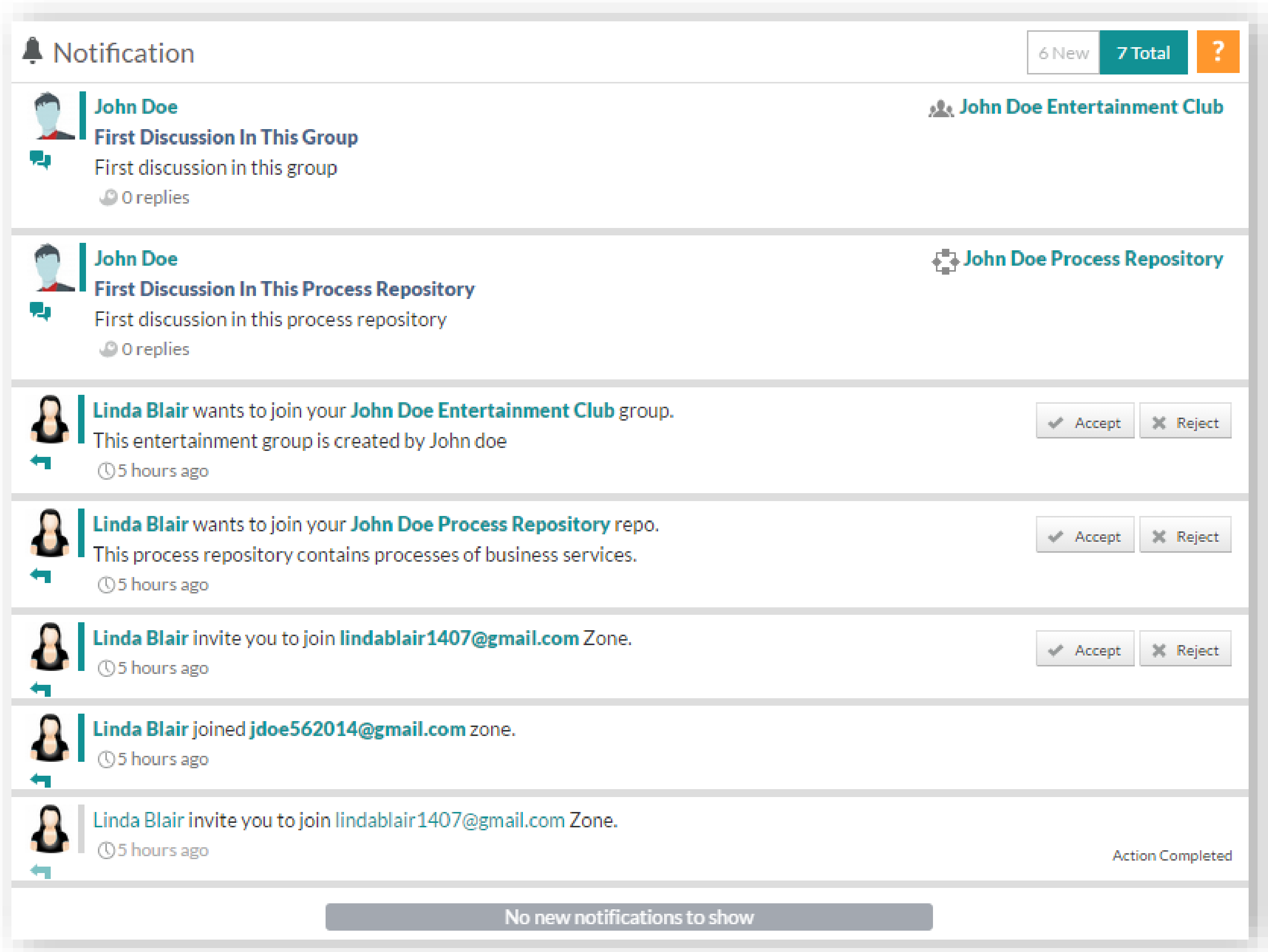
Notification

List down all recent posts and activities of Appiyo members (20 per page) for your view.

To view more notifications click **Show More.**

For unread notifications (highlighted in dark green color), click on the notifications to change the status to read.

Navigation: Home tab



© Copyright 2014 Appiyo. All rights reserved.

Tasks

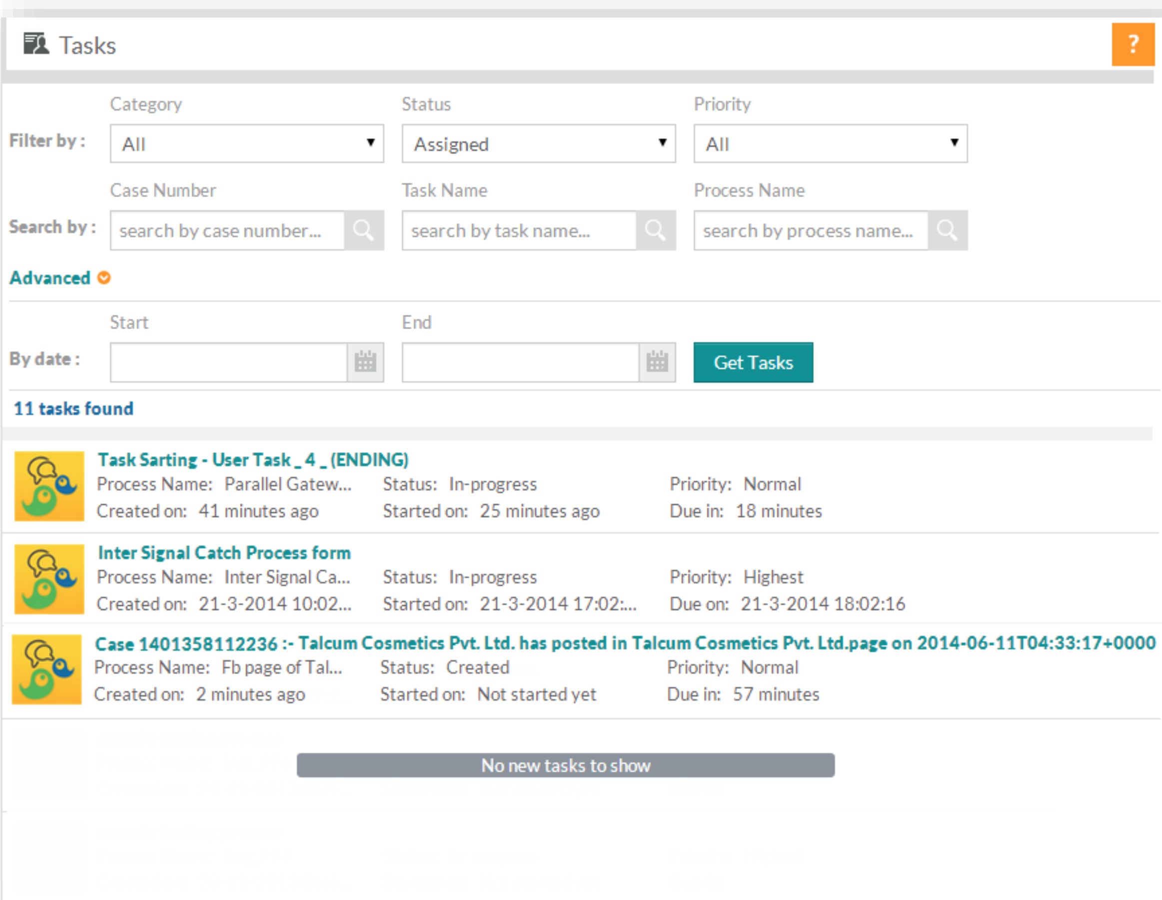
- Every time a workflow is initiated, an explicit task in the workflow generates a unique **instance number**. All tasks in the workflow are collected under an instance number.
    - Instance number is a very useful organizing principle provided in Appiyo architecture to introduce queue analysis and management later.
    - A script task or any other mechanism can generate an instance number.
    - You click on a Task to act on it. Tasks are identified by their names.
    - Task list is also delivered on mobile.

Displays the list of tasks assigned to you and your team.

You can **filter the result** by their **Category** (**All/Assigned to me/Assigned to my team**), **Status** (**Assigned/Completed**) and **Priority** (**All/Lowest/Low/Normal/High/Highest**).

Or **search for tasks** by **Case Number**, **Task Name** and **Process Name** or by **Start** and **End** date (under **Advanced**).

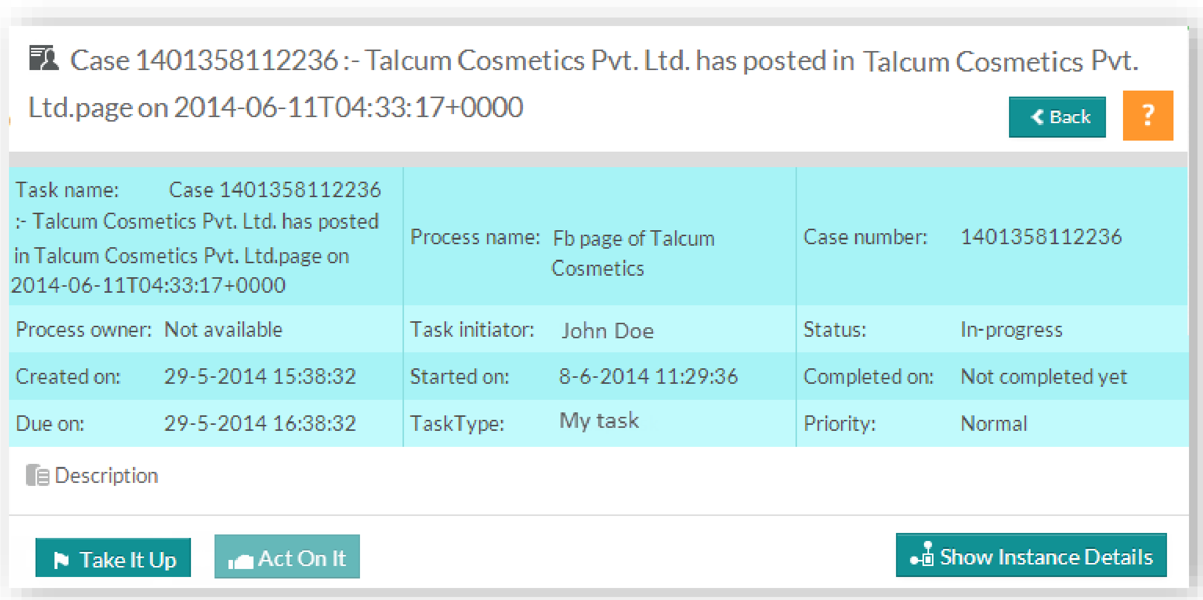
Navigation: Tasks tab



© Copyright 2014 Appiyo. All rights reserved.

You can click on the required task to get its instance details and to take necessary action.

- ‘**Take it up**’ allows a task to be owned by a person intending to execute it.
  - Only after ‘taking up a task’ can you ‘**Act on it**’.
  - After you take up a task for processing, it may so happen that you do not want to process it immediately for some reason. You can ‘**Release it**’. Released tasks can be re-assigned by a supervisor to another member. If the task released is a group task, any other group member can **Claim** it directly.
  - This concept is useful to balance task load through redistribution, if necessary. It will also be useful for tasks assigned to groups wherein any group member can perform the task depending on her/his availability.
- If you are keen to know workflow progress, you may use ‘**Show Instance details**’. It makes the process instance transparent.



Here is where a task gets done. User Task is used (in process) to create and assign tasks. Tasks can be assigned to single user or to a group. The task priority can be set as Normal, High, Highest, Low and Lowest. There are four interesting actions possible here.

© Copyright 2014 Appiyo. All rights reserved.

User assigned task

If a task is assigned to an individual, then three action buttons become relevant - 'Take It Up', 'Release' and 'Act On It'. The task assigned to user will follow task states in the given order:-

Button	Task Status	Description
-	Created	Task has been created and assigned to the user. It is ready to take up and work on.
Take It Up	In-progress	The user is working on the task. The user can either '**Release**' the task or '**Act On It**'.
Release	Created	Releases the task that has already been taken up (moves to initial state).
Act On It	Completed	The user has completed the task.

Team assigned task

If a task is assigned to a group, a group member who is free can take up the task for execution before actually executing it. Here three action buttons become relevant - 'Claim', 'Release' and 'Act On It'. The task assigned to group will follow task states in the given order:-

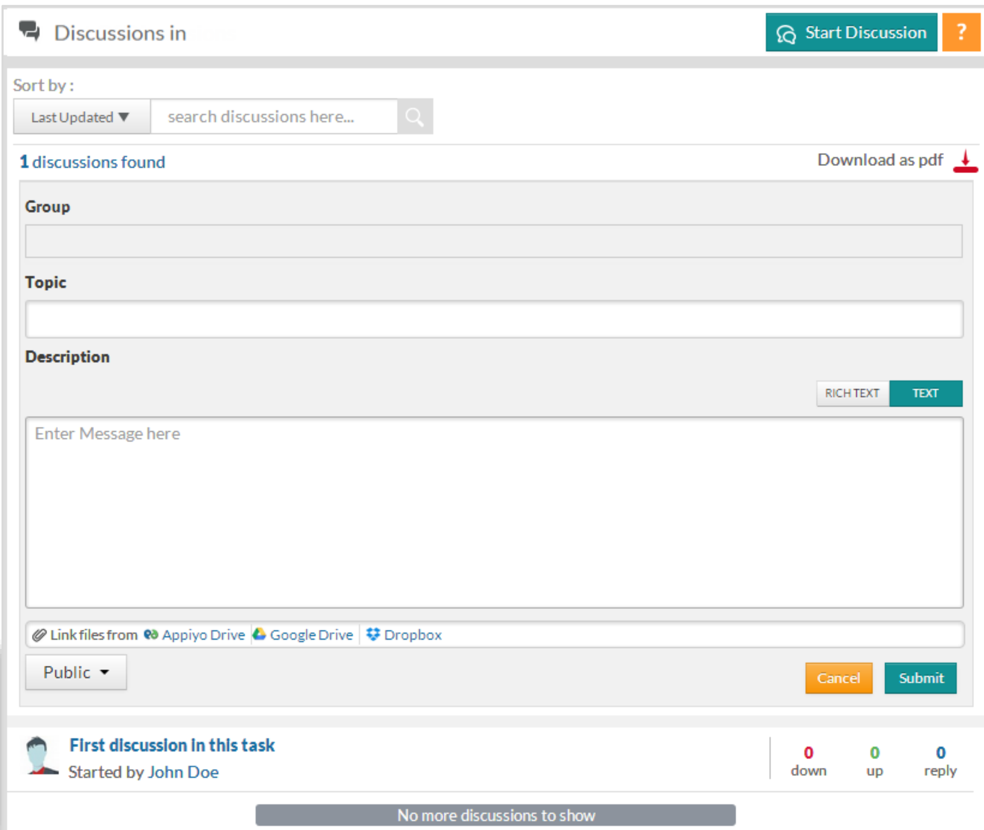
Button	Task Status	Description
-	Ready	Task has been assigned to the group and ready to claim and work on.
Claim	In-progress	The task is reserved by the user of the group and is working on it. The user can either '**Release**' the task or '**Act On It**'. Other Users of the group cannot work on it if not released by the user who has taken up the task.
Release	Ready	Releases the reserved task (moves on to initial state) and allows other users of the group to '**Claim**' it.
Act On It	Completed	The user has completed the task.

© Copyright 2014 Appiyo. All rights reserved.

Discussion

- While you act on a task, you may want to seek explanations or exchange views in that context with your co-workers. It is a task related conversation and you may want to adduce documents, pictures, and links. You can '**Start discussion**' to initiate a conversation.
  - 'Start discussion' is a very useful feature which can eliminate email inconvenience. It can also help you to dig into trails very easily later.
  - By bringing this in conjunction with task execution, Appiyo helps to converge unstructured (engagement) data with structured (transaction record) data. Together they can be subject to big data analysis over time.
  - Being aware of this powerful feature and if applied mindfully, it can prove to be extremely useful in improving overall performance.

Navigation: Required Task → Start Discussion



To **start a discussion** for any task (under **Tasks**), click **Start Discussion** under **Discussion** tab of the required task and enter necessary details under **Group**, **Topic** and **Description**.

For **Description**, you can choose either **RICH TEXT** or **TEXT** option located at right side.

Set the type as **Public** or **Private.** You can also attach your files from **Appiyo Drive**, **Google Drive** or **Dropbox** and click **Submit**.

To **notify** about a discussion to **all** group **members** (under discussion reply), click **Notify all**.

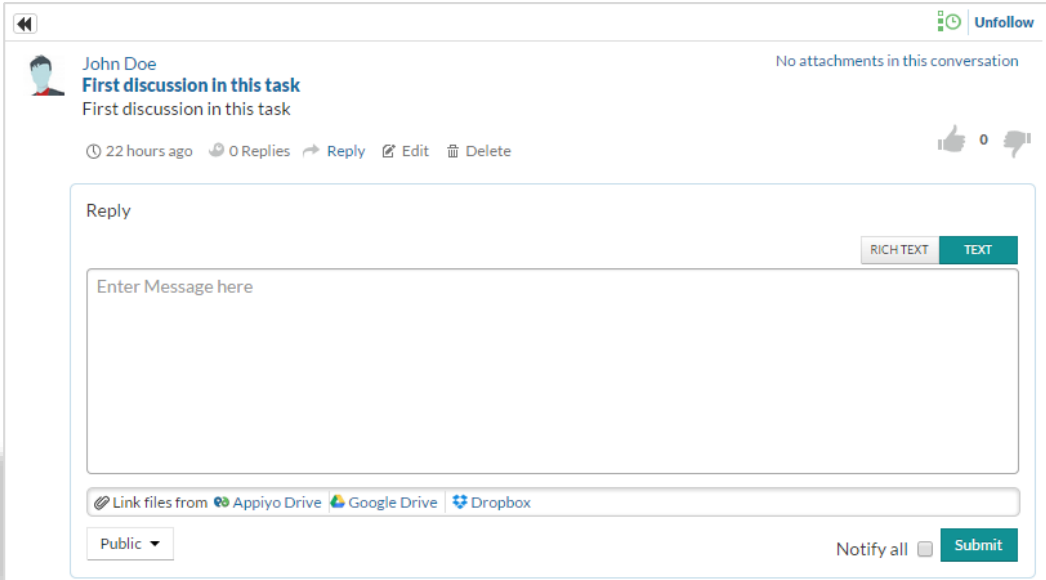
To **notify required members** (under Private discussions), include the members list in the textbox near **Private** button.

© Copyright 2014 Appiyo. All rights reserved.

To **filter discussions**, search for discussions under **search discussions here**. You can also search for discussions by selecting either of the two options **Last Updated** or **Last Created**.

To **download** all the **discussions as a pdf** file, click **Download as pdf**.

To **edit discussion, delete discussion, follow/unfollow discussion** and **up/down vote**, click on the required discussion.



To **modify your discussion**, click **Edit** from the discussion (to be modified), make necessary changes and click **Update**.

To **delete your discussion**, click **Delete** from the discussion (to be deleted) and click **yes**.

To **get notifications about discussions,** click **Follow** of the required discussion.

To **stop getting notifications about discussions,** click **Unfollow** of the required discussion.

To **add your like to discussion** click the **icon** from the required discussion.

To **add your dislike to discussion** click the **icon** from the required discussion.

© Copyright 2014 Appiyo. All rights reserved.

Process

This page is specific to the processes you have created and subscribed. While you use this page for viewing the subscriber list on your process, you can also learn the customer ratings and focus on improving your processes.

The process repositories includes the modules, workflows, forms, datasources, participants and app menus of the business applications.

To **get** the list of **created and/or subscribed process repositories**, click the required option (**All**/**Created**/**Subscribed**) from the drop down list.

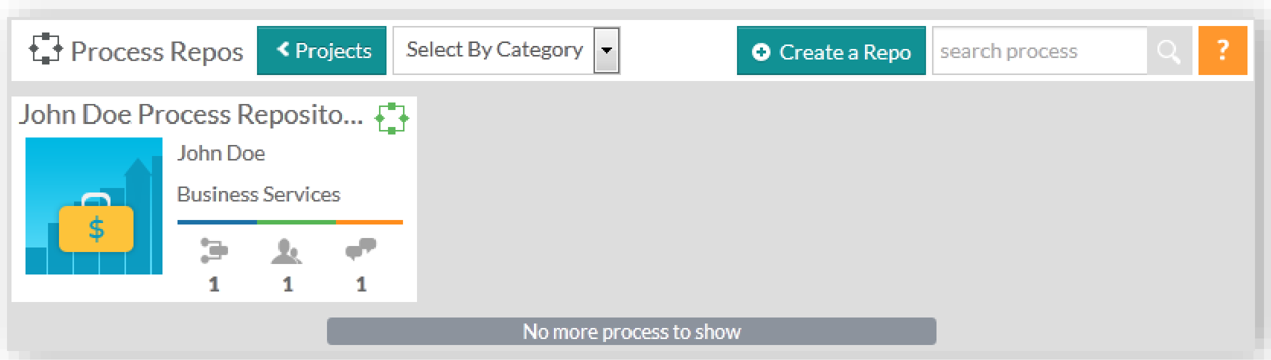
Explore will display all the process repositories available on the zone.

Navigation: Process tab

Filter Process Result

You can filter the process repository result to get the details of any required process repository. You can also filter by their category by searching under **Select By Category**. In process repository, you can modify your project by adding, updating, deleting modules, forms, workflows, datasources, participants and app menus.

To view any process repository, click on the process repository from the list which displays its information, subscribers list, discussions about it, its modules, configuration and app menus.



© Copyright 2014 Appiyo. All rights reserved.

Groups

This page lists the Groups you have created and subscribed.

To **get** the list of **created and/or subscribed groups**, click the required option (**All**/**Created**/**Subscribed**) from the drop down list.

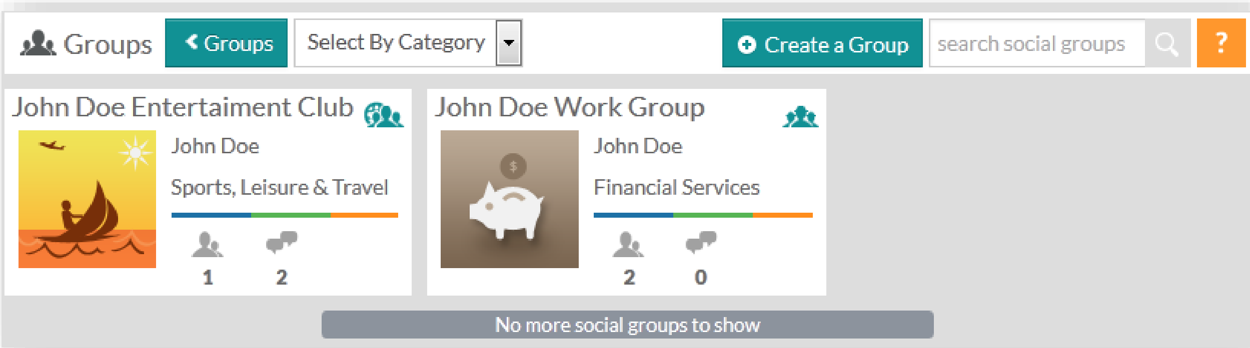
Explore will display all the groups available on the zone and social groups on Appiyo (which are appiyo social groups).

Navigation: Groups tab

Filter Groups Result

You can filter the groups result to get the details of any required group. You can also filter by searching under **Select By Category**.

To view any group, click on the group from the list which displays its information, subscribers list, discussions and cases about it.



© Copyright 2014 Appiyo. All rights reserved.

Applications

This page lists the applications you have created and subscribed.

To **get** the list of **created and/or subscribed applications**, select the required option (**All**/**Created**/**Subscribed**) from the drop down list.

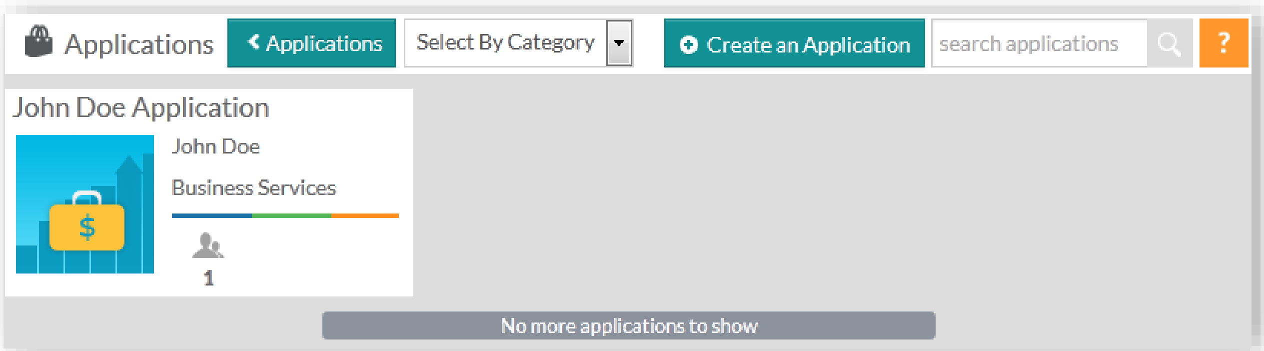
Explore will display all the applications available on Appiyo.

Navigation: Applications tab

Filter Application Result

You can filter the application result to get the details of any required application. You can also filter by searching under **Search By Category**.

To view any application, click on the application from the list which displays its information and subscribers list.



© Copyright 2014 Appiyo. All rights reserved.

Monitor

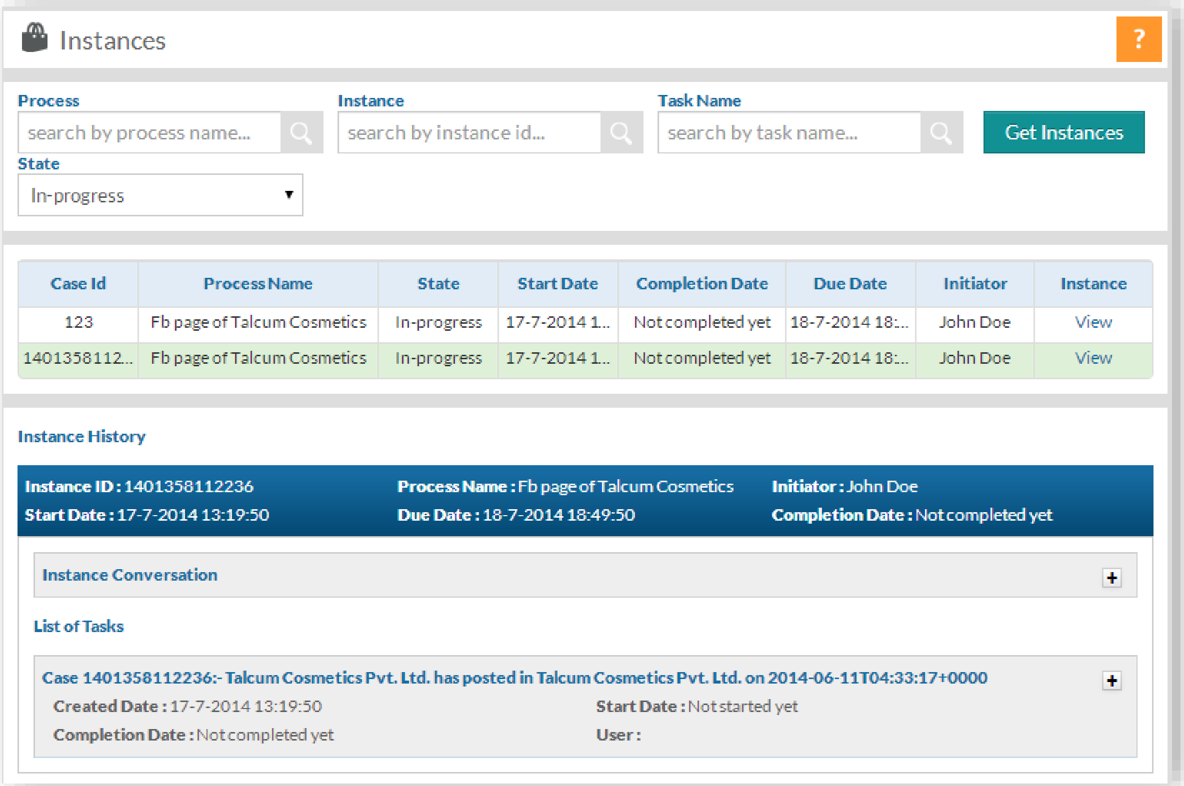
You get a status view into instances of processes here. The essential idea is to enable informed interventional decisions about these entities. For example, analysis of tasks may lead to the discovery of an overloaded employee, prompting additional resource deployment or work re-distribution.

Navigation: Monitor tab

Instance is used to monitor processes (In-progress/Completed) assigned to user. Case Id (instance id) should be generated by the user (refer to User Task) to monitor processes. Process and Sub Process can have many instances.

You can **filter the instance result** by **Process** (process name), **Instance** (instance id), **Task Name** and **State** (**In-progress/Completed**).

You can click on the required instance to get its details.



© Copyright 2014 Appiyo. All rights reserved.

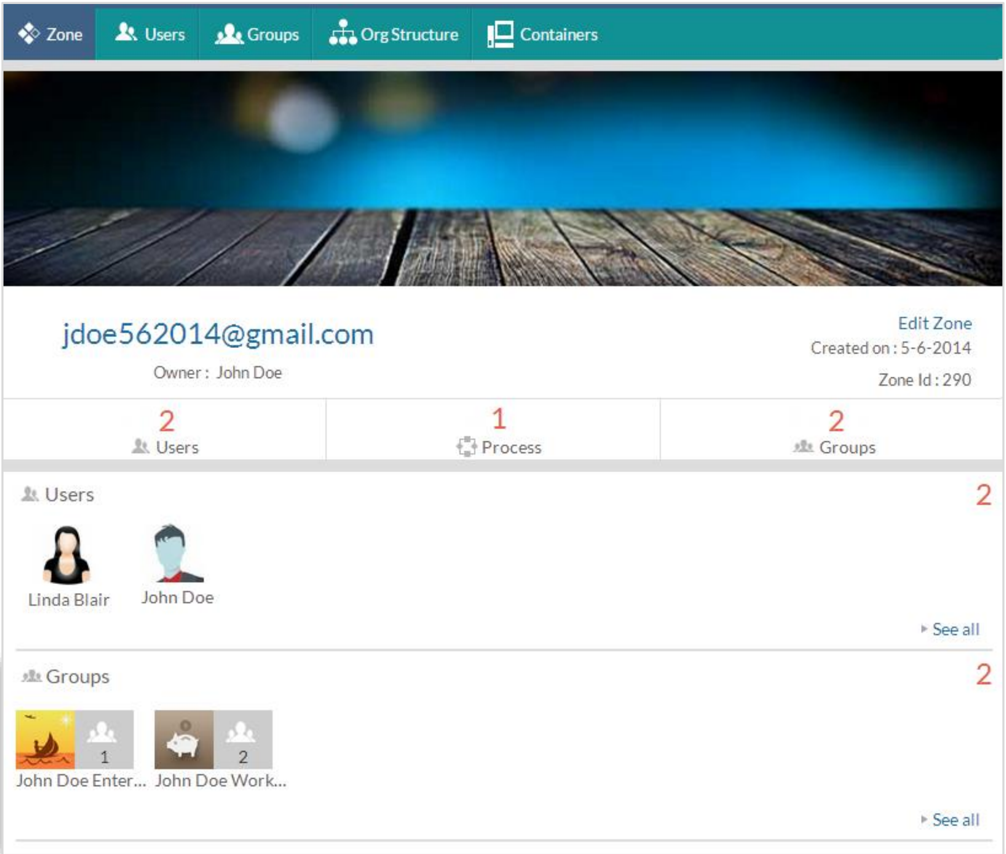
Admin

This page includes all the details about your zone. Zone details include list of your zone users, subscribed and created processes as well as groups, zone (members) organizational structure and your container details. Here the zone admin can add new users or invite existing users to their zone.

Navigation: Admin tab

Zone

This page lists the zone users with the processes and groups you have created and subscribed. Here you can also edit your zone profile and can give zone administrative rights to zone members (refer to **Setup Zone Profile**).



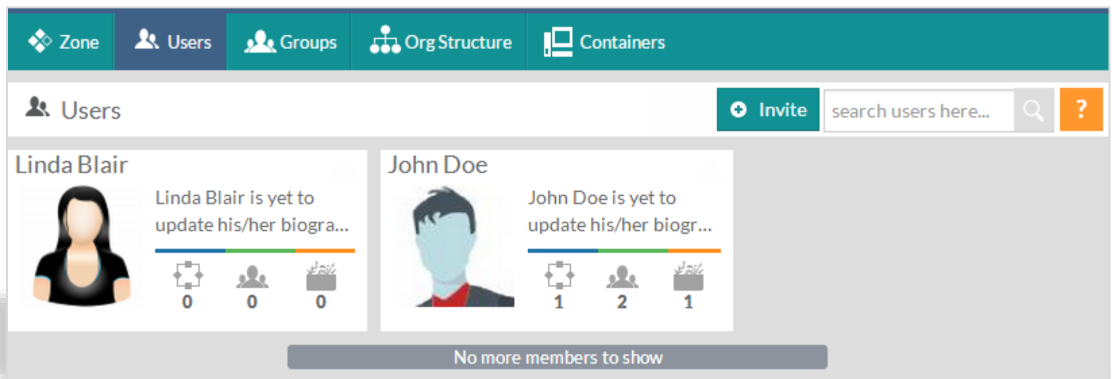
© Copyright 2014 Appiyo. All rights reserved.

Users

- A zone administrator can choose to invite members from outside the zone. Two possibilities exist.
  - The invitee may not yet be an Appiyo member. In this case, the invitee will receive an activation link into her/his email id. Upon activation, the invitee automatically becomes an Appiyo member with a community zone created and allotted and, she/he will also become a member of the inviting zone.
- In the second possibility the invitee is already an Appiyo member. On accepting the invitation, the invitee becomes a member of the inviting zone.

Navigation: Admin tab → Users tab

Displays the list of your zone users. You can also add new users or invite existing users to your zone (refer to My Zone users).



Edit Reporting Manager

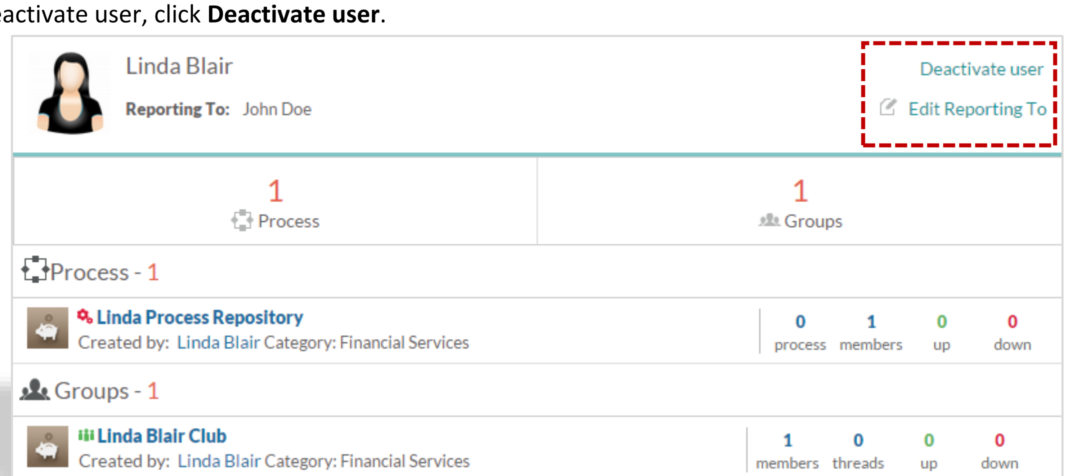
You can edit your zone's reporting manager (only zone users with administrative rights) to whom other users of your zone can report about their tasks.

Click on the required user and **Edit** the **Reporting** manager.

Deactivate user

Deactivate user allows you to delete your zone users from the zone.

To deactivate user, click **Deactivate user**.

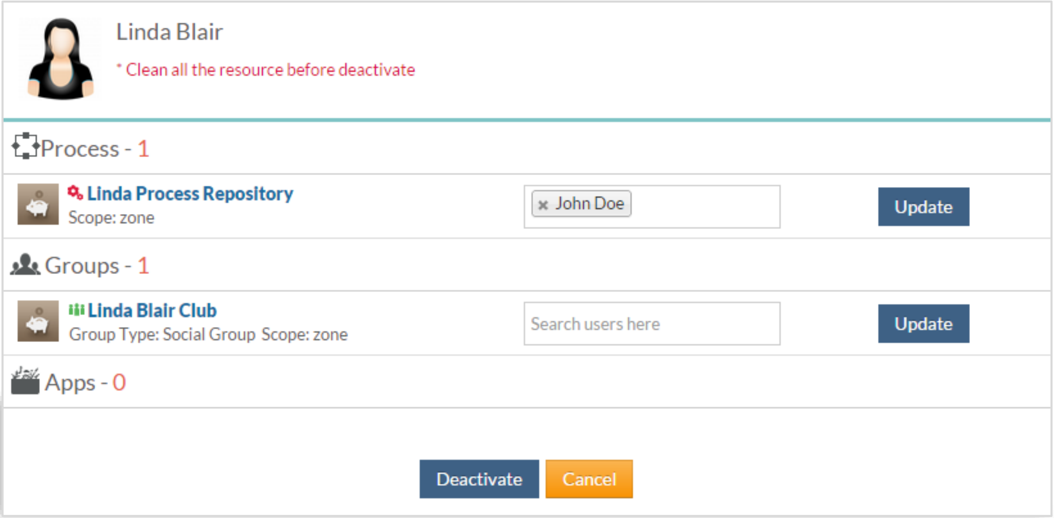


© Copyright 2014 Appiyo. All rights reserved.

Before deactivating the user, you can change the ownership of the user's resource to continue using them in your zone.

To **change ownership** of the **user's resource**, **Search** for **users** (new owner) under required resource and click **Update**.

Click **Deactivate** to delete the user from your zone.



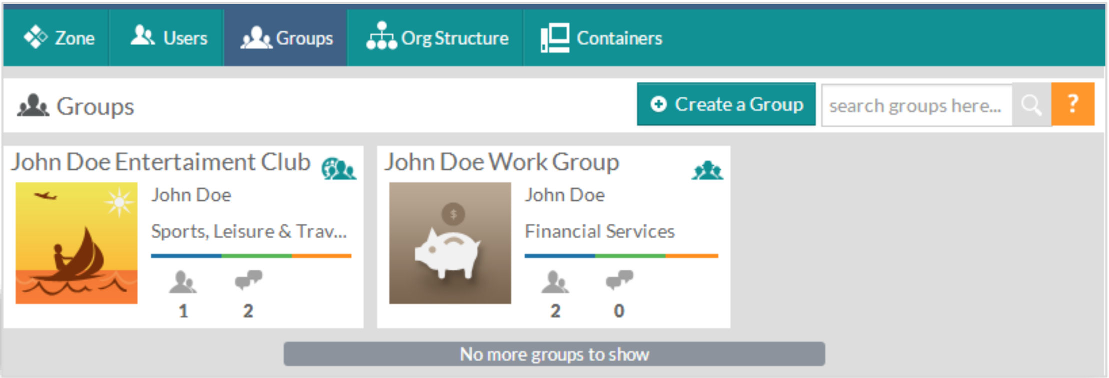
© Copyright 2014 Appiyo. All rights reserved.

Groups

An administrator of a zone can create groups and assign members to the groups. Two types of groups are possible. One is a user group and the other is a workgroup. The latter, as the name suggests, is specifically meant to assign members to workgroups for performance of tasks. The user group is a general purpose body of members. Only a zone administrator can create workgroups within a zone.

Displays the list of groups you have created and subscribed.

Navigation: Admin tab → Groups tab

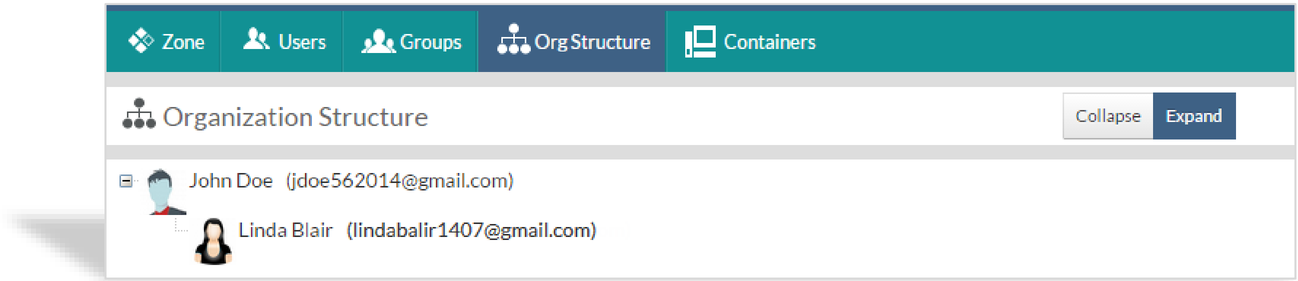


© Copyright 2014 Appiyo. All rights reserved.

OrgStructure

Navigation: Admin tab → OrgStructure tab

Displays the list of zone members prioritized from higher to lower ranking in the organization.



© Copyright 2014 Appiyo. All rights reserved.

Containers

An administrator of a zone can view the containers in her/his zone and their utilization details. Since containers are billable resources in the commercial zone, such a view will give transparency to their billing. This will be useful to manage and optimize the usage of zone containers. For example, a dormant process may be un-deployed to free up container capacity. To help you recollect, a container is a slice of an operating system. Instead of hypervisors, Appiyo uses container virtualization. You may want to refer to our whitepaper on this topic.

A container is a computing resource (virtualized OS). It includes details about your resources and processes. Resources include details about your maximum memory limit, used memory, your container id (user id) and process id and the status. Also provides information about the number of processed and queued messages, created, scheduled and deployed processes.

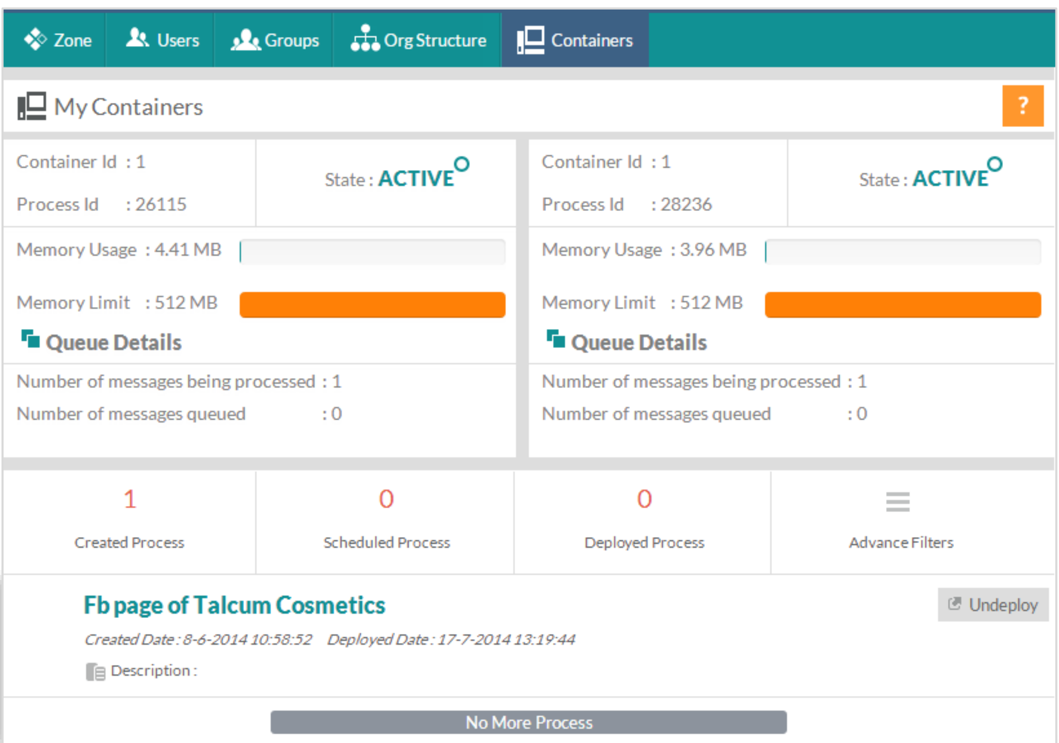
A community zone will be powered and limited by a 128MB container.

A commercial zone will be powered by a 512MB container or multiples of it. It can scale with need.

Navigation: Admin tab → Containers tab

Undeploy

To **Undeploy** the **deployed processes**, click **Undeploy** of the required process.



© Copyright 2014 Appiyo. All rights reserved.

Menu

Recent Workflows	Shows the list of recently visited workflows.
Recent Forms	Shows the list of recently visited forms.
Recent Process (repository)	Shows the list of recently visited process repositories.

© Copyright 2014 Appiyo. All rights reserved.

References

Get Page and Post Id from Facebook

1. Move the cursor over required post on your Facebook page. is shown on the top right corner of the post
2. Click the **button** and click **Embed Post**
3. Opens **Embed this Post** window with post details.
4. Copy the post id from the post detail shown

Ex post detail- <div id="fb-root"></div> <script>(function(d, s, id) { var js, fjs = d.getElementsByTagName(s)[0]; if (d.getElementById(id)) return; js = d.createElement(s); js.id = id; js.src = "//connect.facebook.net/en_GB/all.js#xfbml=1"; fjs.parentNode.insertBefore(js, fjs); }(document, 'script', 'facebook-jssdk'));</script>
<div class="fb-post" data-href="https://www.facebook.com/permalink.php?story_fbid=**601261103323472**&id=**600529310063318**" data-width="466"><div class="fb-xfbml-parse-ignore">Post by My Library.</div></div>

In above example – **601261103323472** is the **Post Id**
600529310063318 is the **Page id**

Charts

Charts is a visual data that represents/provides information in form of table, graph, or diagram. Appiyo supports Single Series and Multi Series charts.

Single Series Charts

Single series chart is a graphical representation that provides information about one item. Following are types of single series charts supported:

- Column 2D/3D
- Bar 2D/3D
- Pie 2D/3D
- Doughnut 2D/3D
- Line
- Area 2D/3D
- Pareto 3D

Multi Series Chart

Multi series chart similar to single series chart is a graphical representation that provides information about multiple items. They are used to compare multiple set of data that belongs to same category. Following are types of multi series charts supported:

- MS Column 2D/3D
- MS Bar 2D/3D
- MS Line
- MS Area
- Stacked Column 2D/3D
- Stacked Area 2D
- Stacked Bar 2D/3D
- Marimekko

© Copyright 2014 Appiyo. All rights reserved.

Chart Examples

Column Chart

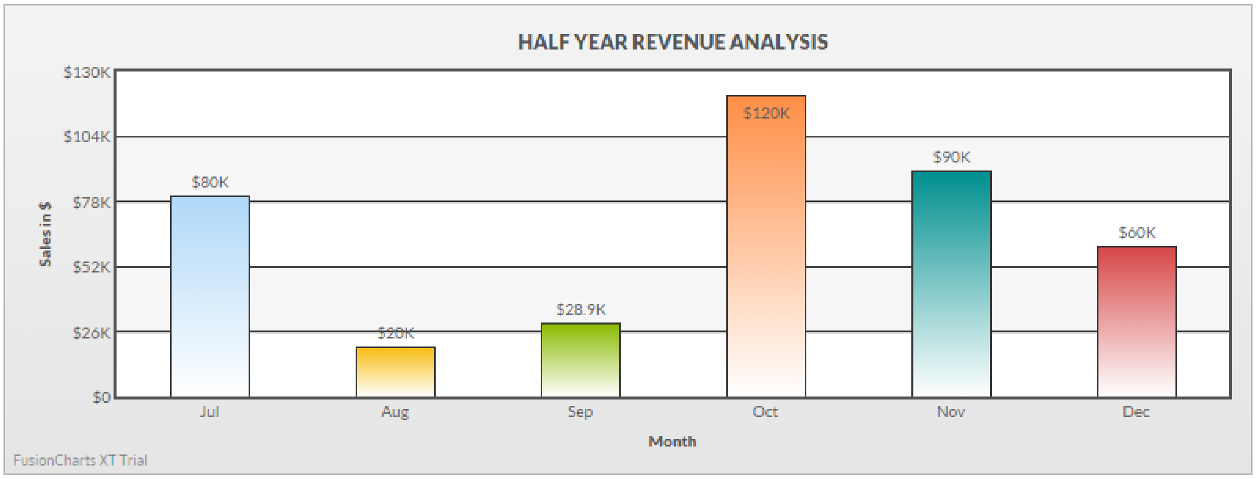
In column chart data are represented by rectangular vertical bars (i.e. vertical bar charts). The length of each of the rectangular bar is proportional to the value it represents.

Usage

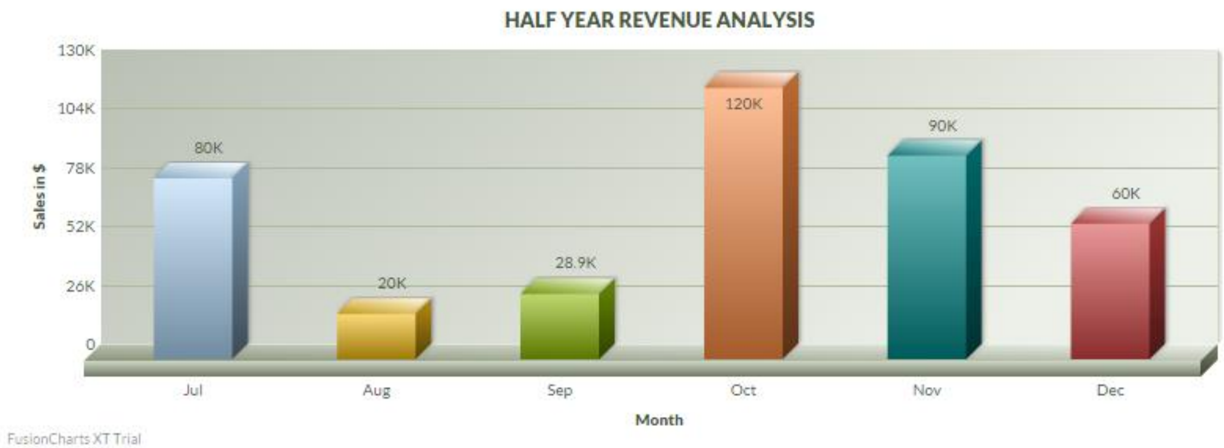
- To compare one or more data
- To show data change over discrete time period (monthly or yearly)
- To show information observed under different situations

Example

A company's half year revenue analysis report between the months July and December (single series data) can be shown using both Column and Bar Chart.



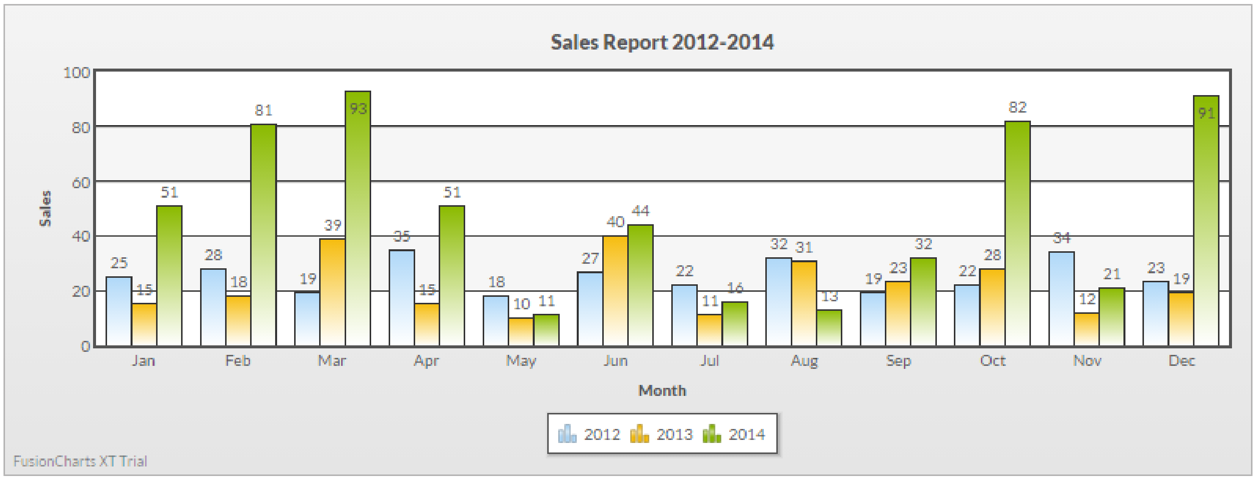
Single Series Column 2D Chart



Single Series Column 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

A company's sales report for the year 2012, 2013 and 2014 is compared using column and bar chart. Since multiple data series (sales report of 2012 - 2014) has to be compared, multi series chart is used here.



Multi Series Column 2D Chart

Multi Series Column 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Bar Chart

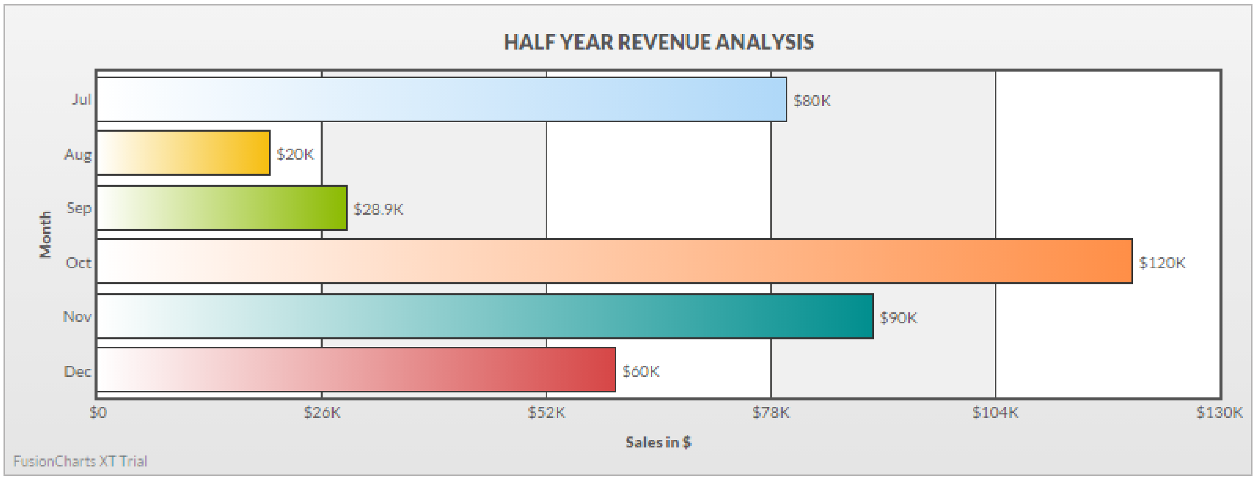
Bar chart similar to column chart is represented by rectangular bars with the x-axis and y-axis interchanged. One axis shows a category to be compared (Month) and the other shows their respective discrete values (Sales in $).

Usage

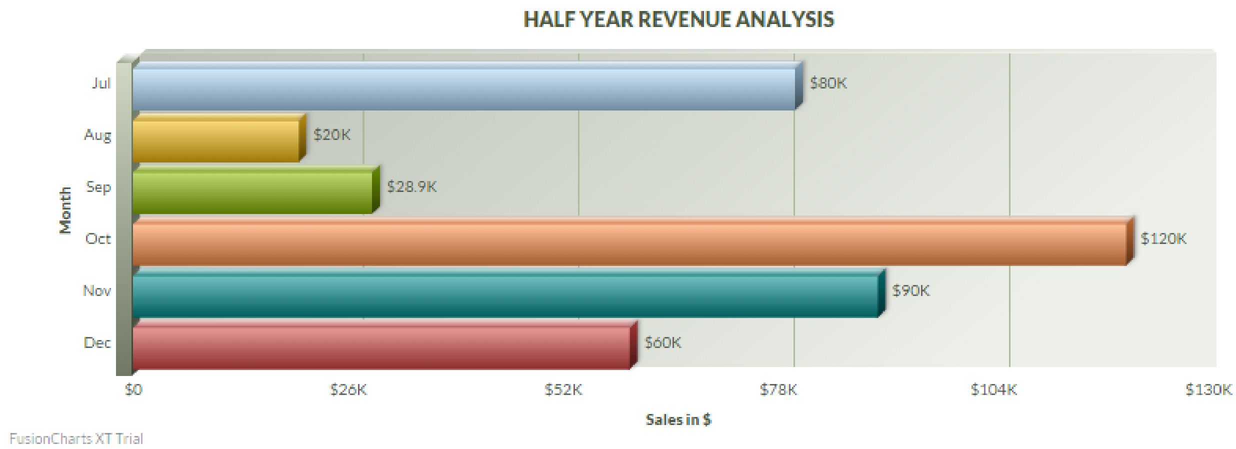
- To show data change over a period of time in descending order
- To fit large data of x-axis (column charts have limited space)
- To compare one or more data

Example

The example half year revenue analysis report of a company is shown in bar chart with the x-axis (Month) and y-axis (Sales in $) interchanged.



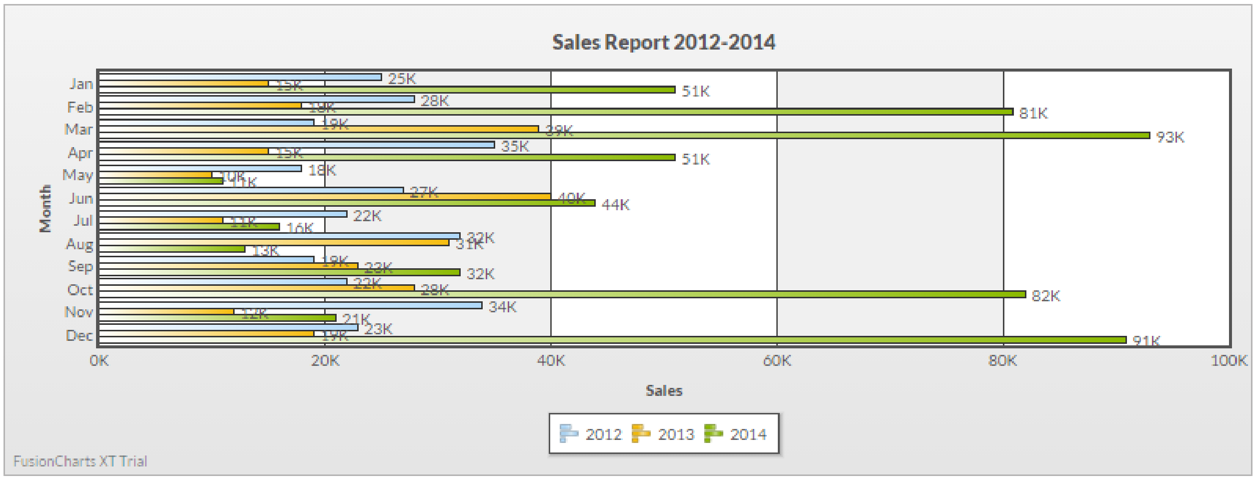
Single Series Bar 2D Chart



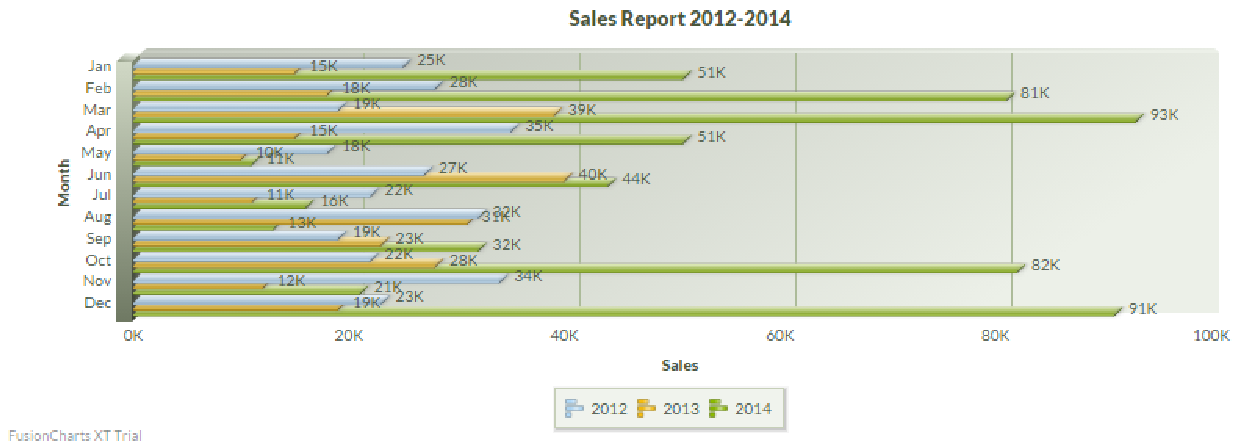
Single Series Bar 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

The example sales report 2012-2014 is shown in bar chart with the x-axis (Month) and y-axis (Sales) interchanged.



Multi Series Bar 2D Chart



Multi Series Bar 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Pie Chart

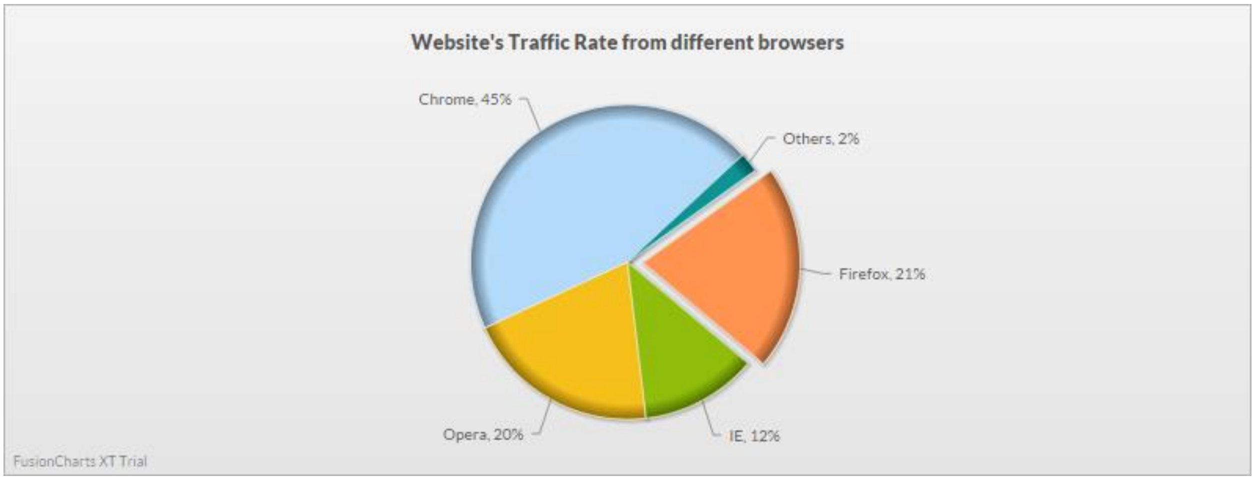
Pie chart is a circular chart divided into sectors with each sector representing data that is proportional to the whole quantity.

Usage

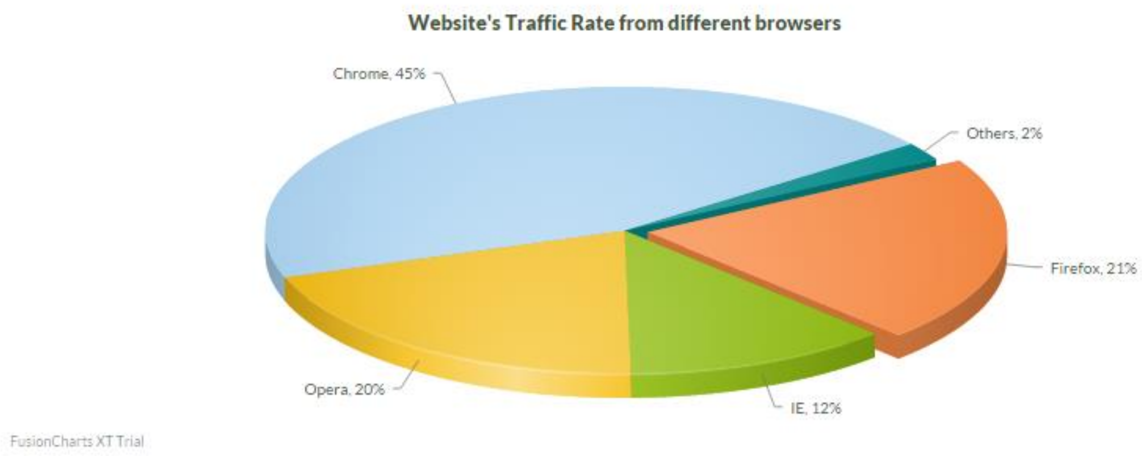
- To show proportional relations of data at a time period
- To compare data parts to whole at a time period
- To show data in percentage
- When the categories (data parts), do not exceed seven
- To show single series data that does not include negative or zero value

Example

For example a website's traffic rate from different browsers can be shown with Pie and Doughnut chart.



Pie 2D Chart



Pie 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Doughnut Chart

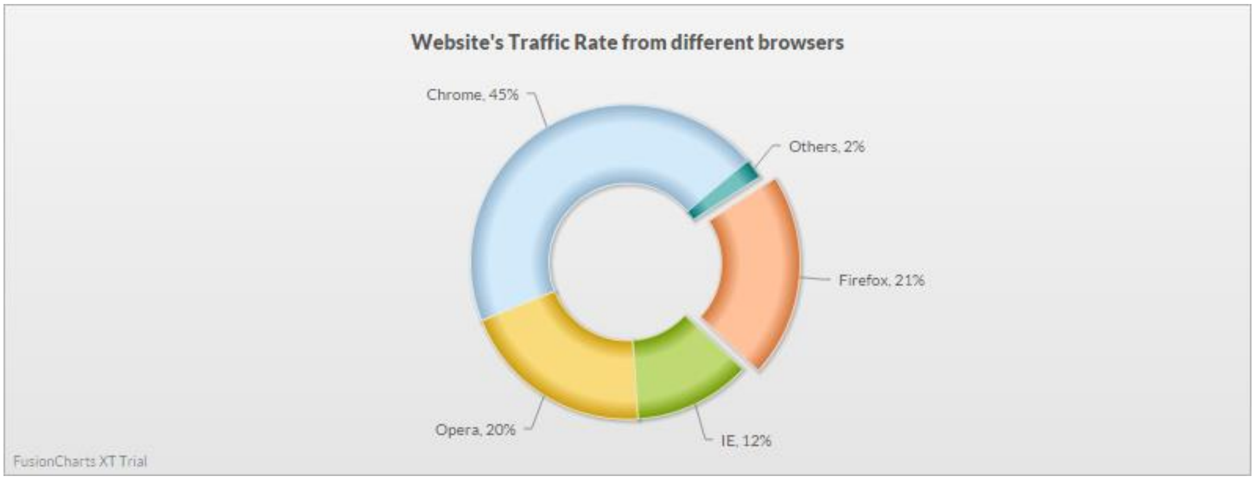
Doughnut chart is similar to a pie chart with a hollow in the center giving the chart a doughnut shape. The data is represented as a percentage of the whole.

Usage

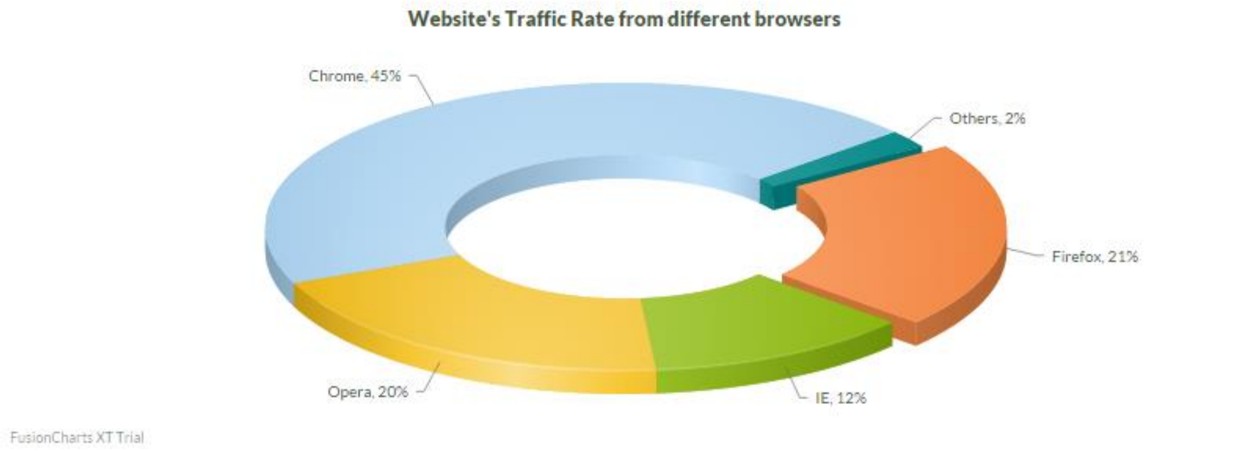
- To show one or more data series as a percentage of the whole
- To show proportional relations of data
- When the categories (data parts in each ring), do not exceed seven
- To show data that does not include negative or zero value

Example

A browser's traffic rate to a website in proportional to the complete traffic rate is shown in the following doughnut charts.



Doughnut 2D Chart



Doughnut 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Line Chart

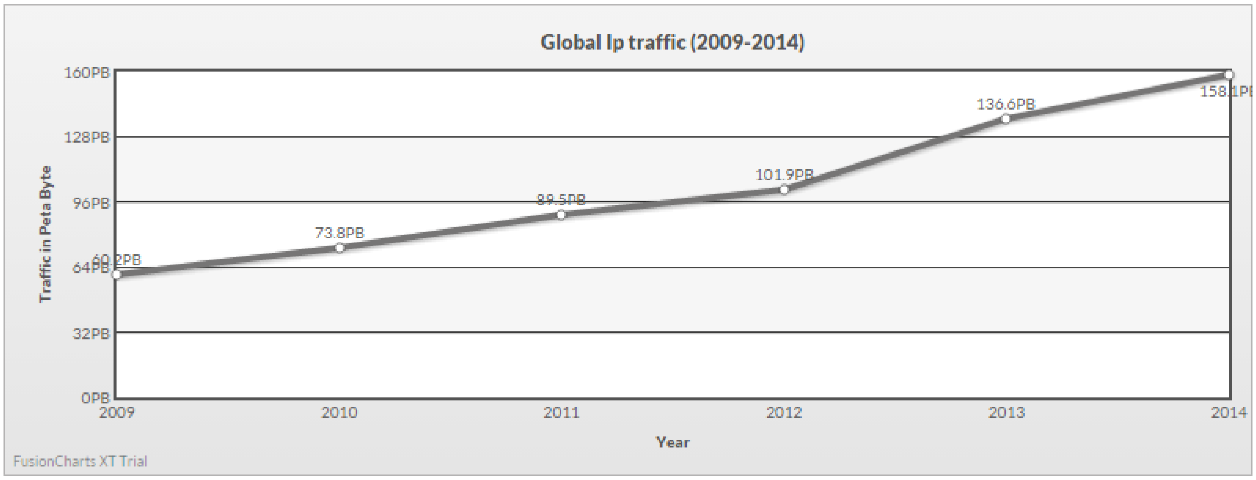
Line charts are used to depict data that changes over time with the series of datapoints (data represented as points) connected by a straight line.

Usage

- To show continuous data change (trend) at regular time interval
- When the category (x-axis) has evenly spaced text or numeric label like months, quarters, years
- When the data change to be compared is large

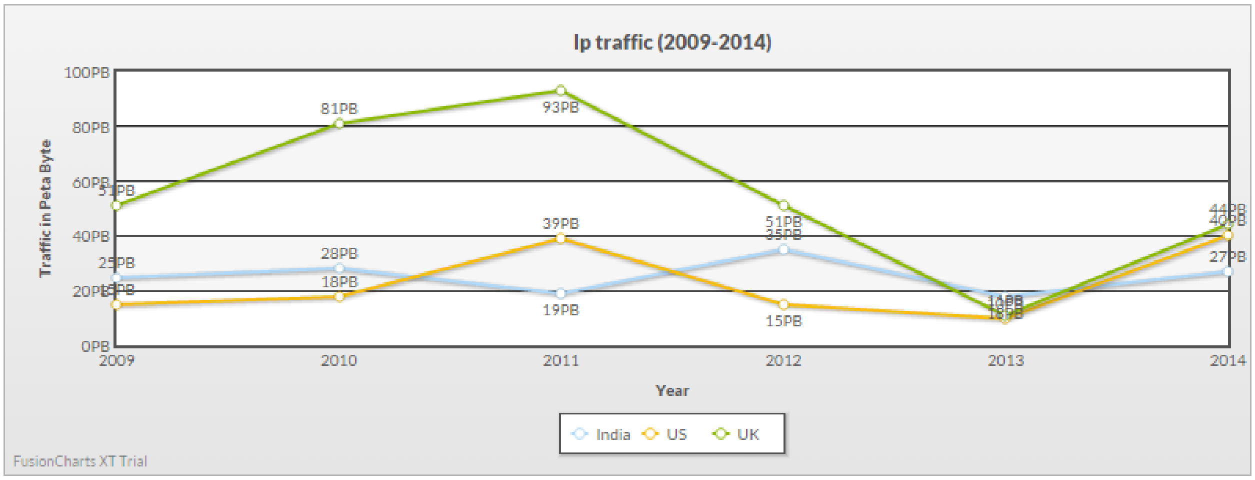
Example

IP traffic rate for the year 2009 – 2014 throughout the globe can be shown using line chart. Since the data is very large bar chart cannot be used to compare the data instead line chart is used.



Single Series Line Chart

In above example, only one data series (global Ip traffic) is shown. But to depict multiple data series (Ip traffic of three places) you would use multi series line chart.



Multi Series Line Chart

© Copyright 2014 Appiyo. All rights reserved.

Area Chart

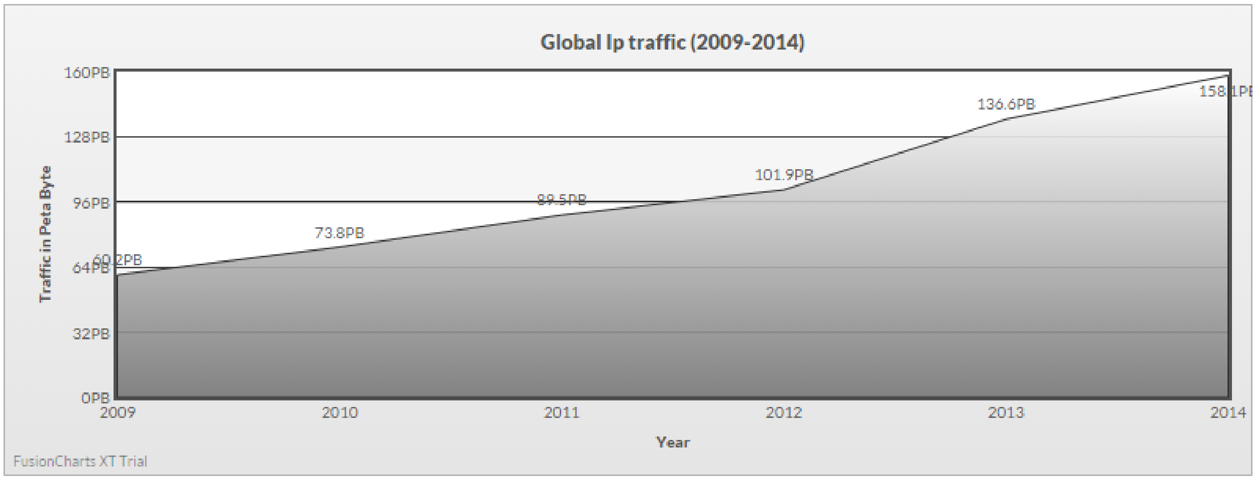
Area chart depicts the quantitive data. It is similar to line chart except that the area between the line (representing data) and the threshold (0 by default) are filled with colors or textures.

Usage

- To show trends over time intervals among related data
- To show relationship of data parts to the whole over time

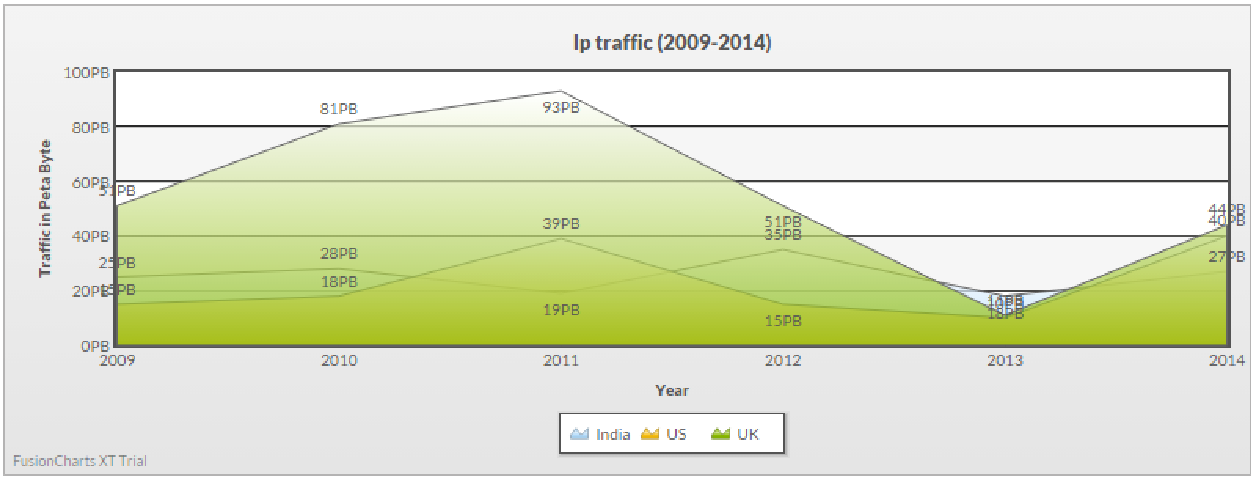
Example

Above same example global Ip traffic can be shown using area chart with datapoints connected by a line along with the area filled with color below the line.



Single Series Area Chart

Ip traffic of three places (multiple data) between the years 2009 – 2014 is shown in the multi series area chart below.



Multi Series Area Chart

© Copyright 2014 Appiyo. All rights reserved.

Pareto Chart

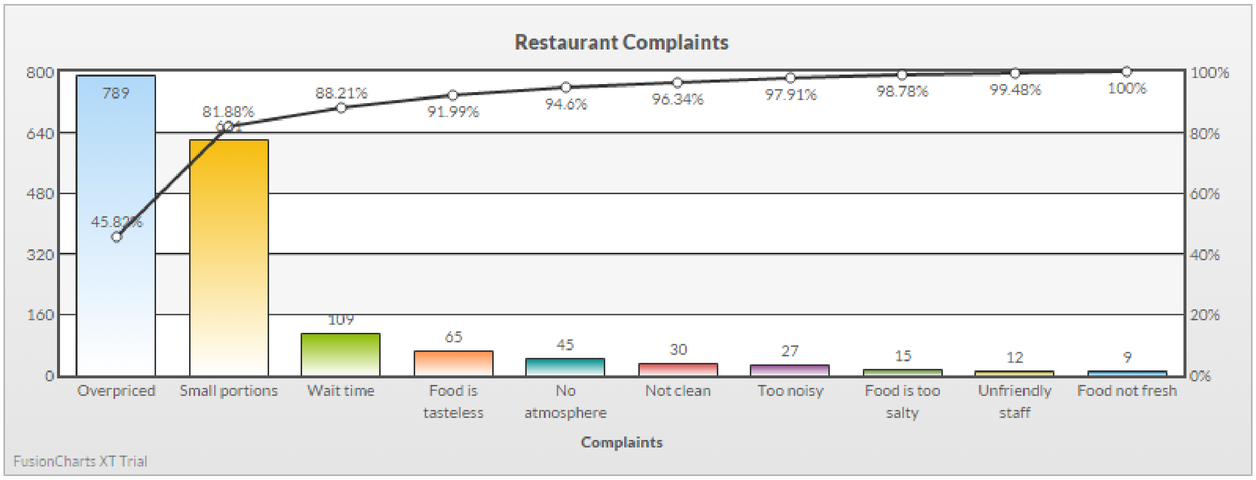
Pareto chart is a combination of both bars and line graphs where data are represented by the bars and their corresponding cumulative total are represented by the line graph. The data for the line graph (i.e. cumulative sum) is calculated automatically with the values given for the bars. It is based on Pareto principle which states 80% of issues are due to 20% of cause. They are the bar charts that are arranged in descending order.

Usage

- To summarize and highlight the important factors of groups of data
- Mostly used to show the issues and their causes
- To prioritize the issues and focus on the most important that needs immediate attention

Example

A restaurant's complaint rate is shown using Pareto chart where the complaints are represented by the bars and their cumulative total (automatically calculated) are represented using line.



Single Series Pareto Chart

© Copyright 2014 Appiyo. All rights reserved.

Stacked Column Chart

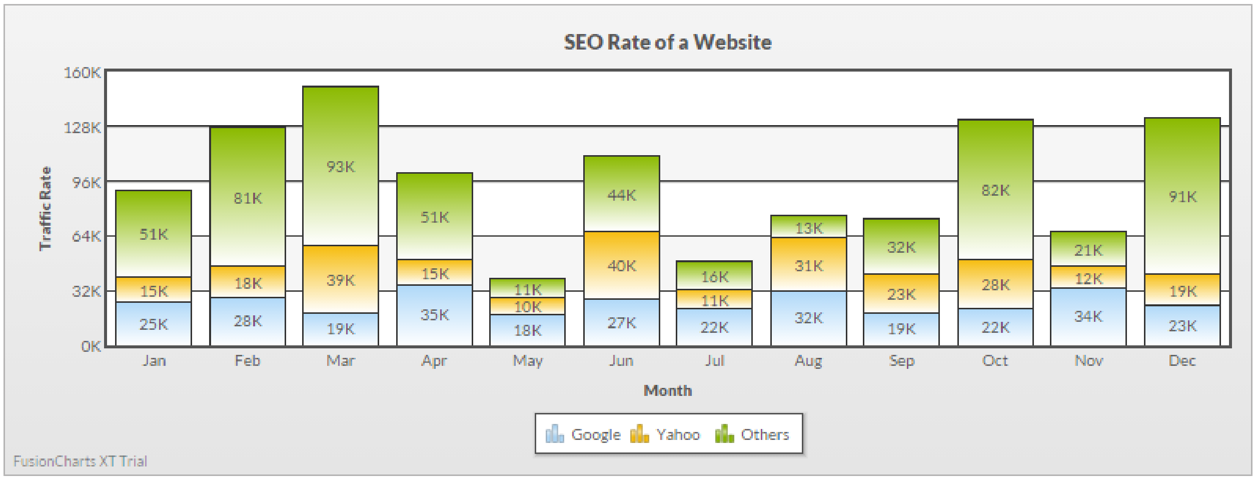
Stacked column chart is represented by vertical rectangular bars with each bar divided into categories (data parts).

Usage

- To compare multi series data and to emphasize their total
- To compare data parts with each other as well as to the whole quantity (each bar represents a whole)
- To highlight the changes between set of data

Example

To see the SEO (Search Engine Optimization) rate of a website its monthly traffic rate from different search engines are depicted in column/bar chart. But if you want to show the traffic rate per month along with the rate from each search engine then stacked charts can be used.



Stacked Column 2D Chart

Stacked Column 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Stacked Area Chart

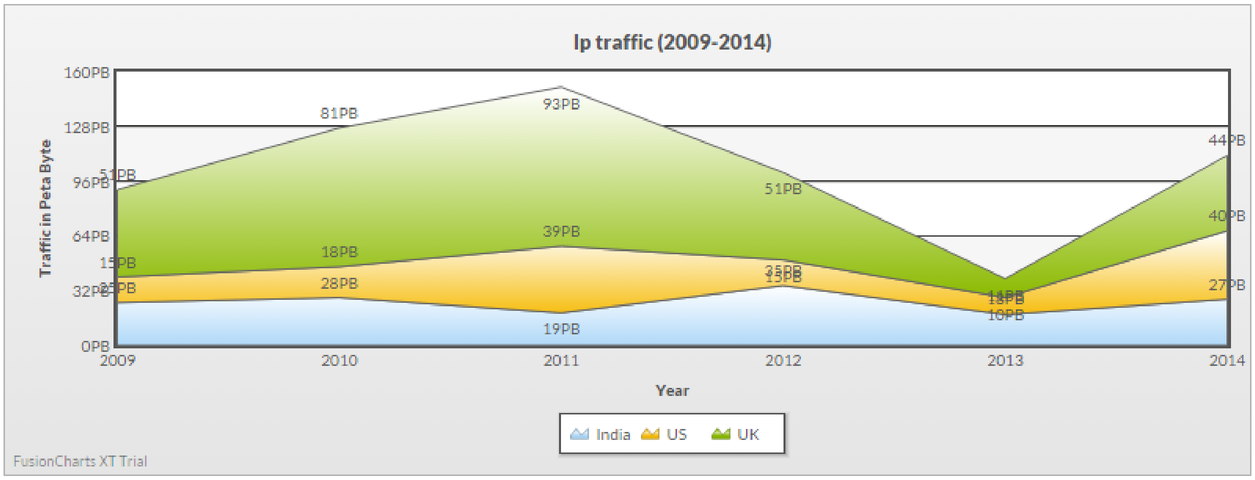
Stacked area charts depicts cumulated totals (of multiple items) that changes over time with the area below the line (representing data) filled with colors or textures. Area chart becomes stacked area chart when the data are stacked on top of other to sum up their cumulative total.

Usage

- To show relationship of data parts to the whole over time along with their cumulative total
- To visualize multiple data set clearly (the upper layer will hide the lower layers in area chart)

Example

As an example, Ip traffic of three places (multiple data) between the years 2009 – 2014 is depicted in the area chart below along with their cumulative totals shown on the right axis.



Stacked Area 2D Chart

© Copyright 2014 Appiyo. All rights reserved.

Stacked Bar Chart

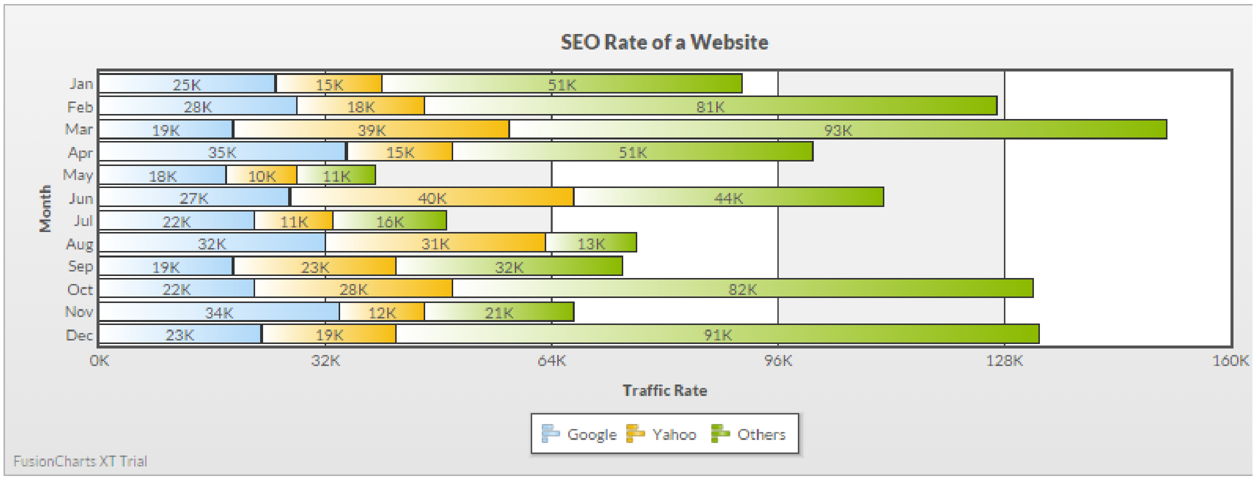
Stacked bar chart works similar to stacked column chart except that the x-axis and y-axis are interchanged.

Usage

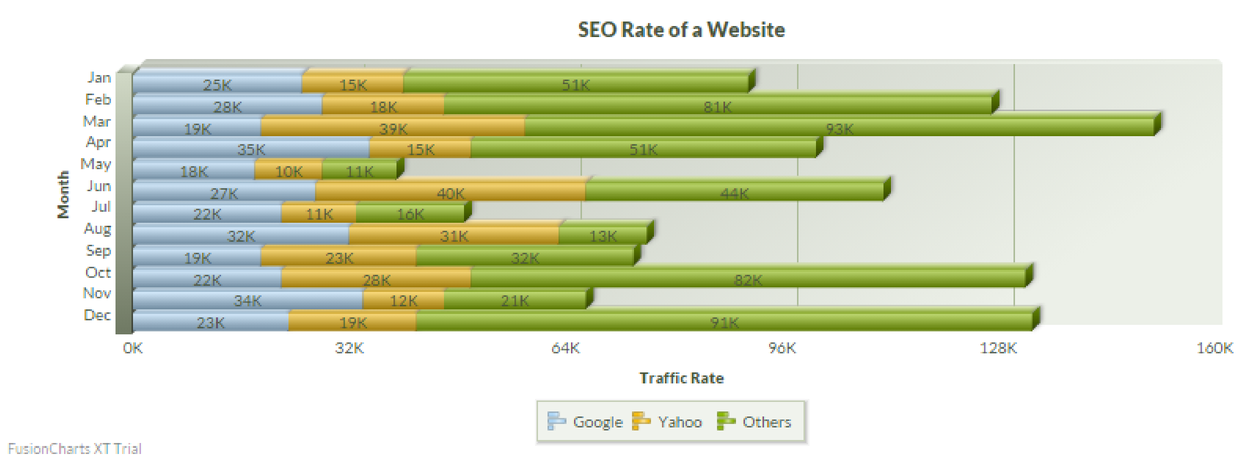
- To show comparison between complex data set
- To highlight the changes between discrete data
- To show relation between data parts and the whole quantity

Example

Example SEO rate of a website is shown using stacked bar chart with x-axis (Month) and y-axis (Traffic Rate) interchanged.



Stacked Bar 2D Chart



Stacked Bar 3D Chart

© Copyright 2014 Appiyo. All rights reserved.

Marimekko Chart

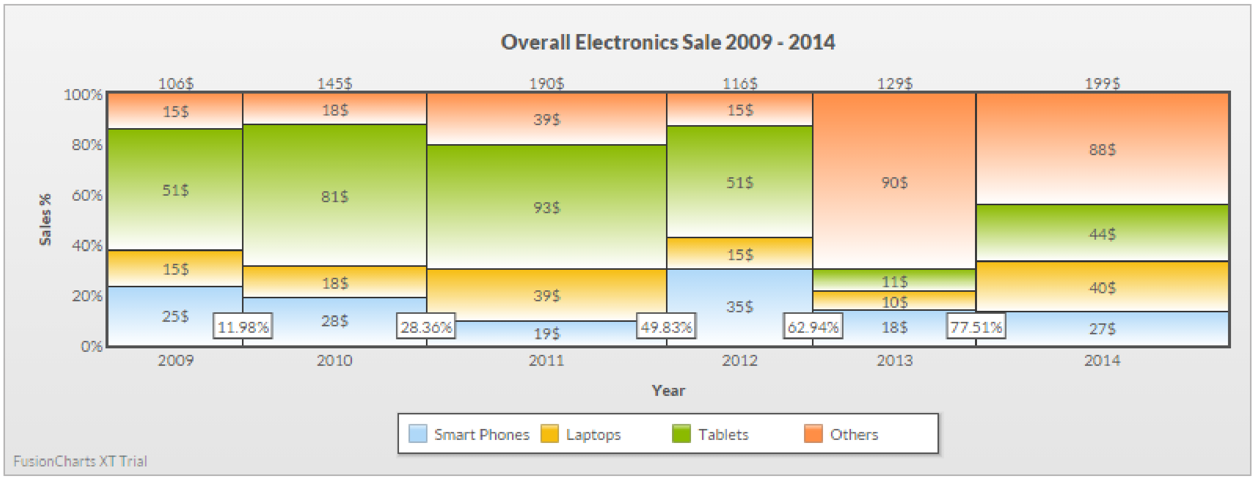
Marimekko chart is a two-dimensional stacked chart (both column and bar). In addition to the varying segment heights of a regular stacked chart, it also has varying column widths. The bars in the chart will be of same height without space between them and in turn are divided into categories.

Usage

- Mostly used for analyzing data related to marketing or sales
- To show quantitative data in two dimension
- To show stacked bars with an additional dimension that has variable width
- To highlight the inefficiencies in an organization

Example

Following example shows the overall electronics sale of a company between the years 2009 – 2014 for different categories.



Marimekko Chart

© Copyright 2014 Appiyo. All rights reserved.

appiyo

Registered Office: 143, Cecil Street, 16-04, GB Building, Singapore 069542

contact@appiyo.com

www.appiyo.com | Follow us on