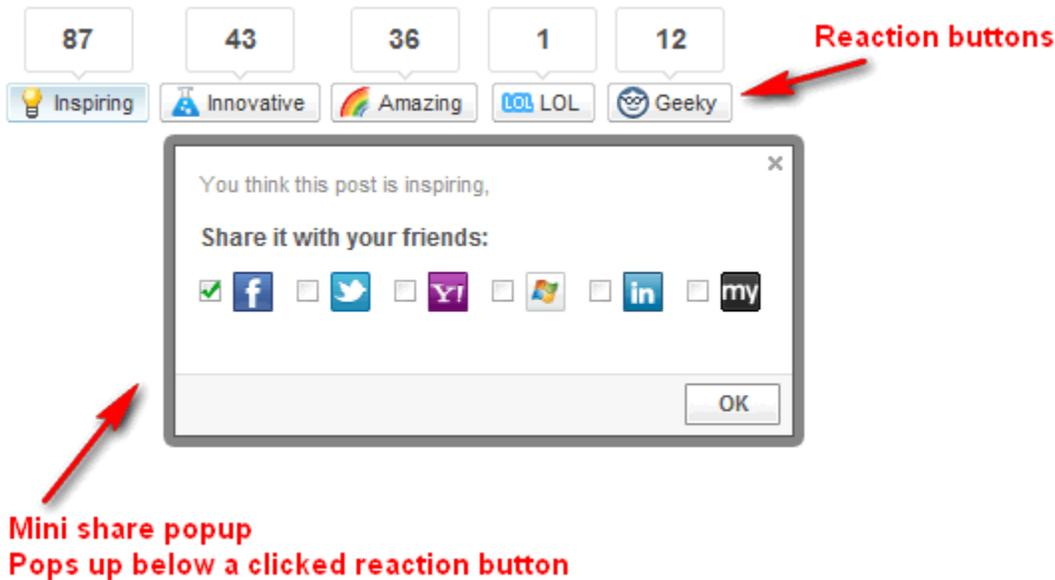


# Reactions

## Introduction

The Reactions add-on allows users to react to content in your site and share their reaction to social networks. The Reactions add-on is built out of one or more buttons, arranged horizontally in a bar. Each button represents a "reaction" to the content, such as 'LOL', 'Amazing', etc. Once a user clicks on a reaction button, a mini Share add-on will pop up to suggest to him to share his reaction to the social networks. The user is given an option to choose to which social networks he wants to post the content. When the user presses the "OK" button, Gigya will publish the user's reaction to each of the social networks that the user selected. The reaction buttons may be displayed with or without counters, representing the number of times a certain reaction to an item was clicked.

You can use the [Reactions Wizard](#) to customize the visual aspects of the Reactions add-on design, view, and grab the customized code.



## What's in this Guide?

This document is a step-by-step guide for integrating the Reactions add-on and also provides a reference and guide for the various customization options. For convenience, we have divided this document into levels of complexity:

- [Quick Start Implementation](#) - learn how to quickly integrate the Reactions add-on and publish user's reaction to the various social networks.
- [Ready-made buttons](#) - Gigya offer a selection of ready-made reaction buttons. Choose your preferred buttons, grab the code and add it to your Reactions add-on.
- [Design your own buttons](#) - learn how to customize the default design of the buttons in the reactions bar.
- [Advanced Sharing](#) - learn how to implement the full richness of newsfeed reports, including adding various media items (Images/ a Flash file/ Mp3 file/ a Video).
- [Automatic Sharing](#) - Learn how to activate an automatic sharing on a reaction in your site without prompting the user with the mini Share add-on.

## Quick Start Implementation

**Before you begin:** Make sure that you have already set up social network application settings before you continue. For more information on how to set up an external application for use with your Gigya account, see the [Setting Up External Applications in Social Networks](#) documentation and select a specific network.

## Implementation Overview

Adding the Reactions add-on is simple, and involves only adding a few lines of code:

- **Step 1:** Define an array of [Reaction objects](#), representing the buttons to display in the reactions bar.
- **Step 2:** Construct a [UserAction object](#), which represents the action taken by a user on your site. The newsfeed that is published to the social networks is constructed from the fields of this object. All the fields of this object are optional.
- **Step 3:** Define the Reactions add-on's parameters.
- **Step 4:** Call the [socialize.showReactionsBarUI](#) API to activate the Reactions add-on.

## Code Example

Add the following code snip to the of your page to activate the Reactions add-on that is presented in the screenshot above:

```
<DIV id=reactionsDiv></DIV> <!-- Reactions add-on DIV Container -->

<script language="javascript">
  // Step 1: Define an array of reaction objects
  var textReactions=[
  {
    text: 'Inspiring',
    ID: 'inspiring',

    iconImgUp:'https://cdns.gigya.com/gs/i/reactions/icons/Inspiring_Icon_Up.png',

    iconImgOver:'https://cdns.gigya.com/gs/i/reactions/icons/Inspiring_Icon_Down.png',
    tooltip:'This item is inspiring',
    feedMessage: 'Inspiring!',
    headerText:'You think this post is inspiring,'
  }
  ,{
    text: 'Innovative',
    ID: 'innovative',

    iconImgUp:'https://cdns.gigya.com/gs/i/reactions/icons/Innovative_Icon_Up.png',

    iconImgOver:'https://cdns.gigya.com/gs/i/reactions/icons/Innovative_Icon_Down.png',
    tooltip:'This item is innovative',
    feedMessage: 'This is innovative',
    headerText:'You think this post is innovative,'
  }
  ,{
    text: 'Amazing',
    ID: 'amazing',

    iconImgUp:'https://cdns.gigya.com/gs/i/reactions/icons/Amazing_Icon_Up.png',

    iconImgOver:'https://cdns.gigya.com/gs/i/reactions/icons/Amazing_Icon_Down.png',
    tooltip:'This item is amazing',
    feedMessage: 'This is amazing!',
    headerText:'You think this post is amazing,'
  }
  ]
</script>
```

```

    }
    ,{
        text: 'LOL',
        ID: 'lol',

        iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/LOL_Icon_Up.png',

        iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/LOL_Icon_Down.png',
        tooltip: 'This made me LOL',
        feedMessage: 'LOL!',
        headerText: 'This post made you LOL,'
    }
    ,{
        text: 'Geeky',
        ID: 'geeky',

        iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/Geeky_Icon_Up.png',

        iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/Geeky_Icon_Down.png',
        tooltip: 'This item is geeky',
        disabledTooltip: 'You found this geeky',
        feedMessage: 'This is geeky!',
        headerText: 'You think this is geeky,'
    }
};

// Step 2: Define UserAction object and fill it with data
var defaultUserAction = new gigya.socialize.UserAction();
defaultUserAction.setTitle("This is the post title");
defaultUserAction.setLinkBack("http://www.gigya.com");

// Step 3: Define the Reactions add-on's params object
var reactionParams = {
    barID: 'today-story', // Identifier of the content to which this
reaction refers
    containerID: 'reactionsDiv', // Reactions add-on DIV Container
    reactions: textReactions, // The reaction array from Step 1
    userAction: defaultUserAction, // The UserAction object from Step
2
    bodyText: 'Share it with your friends:', // optional - text that
appears in the Share popup
    showCounts: 'top', // optional - displays the counters on top of
the buttons
    cid: ''
};

// Step 4: Load the Reactions add-on:

```

```
gigya.socialize.showReactionsBarUI(reactionParams);  
</script>
```

## Code Explanation

### Step 1: Define an array of Reaction objects

Define an array of `Reaction` objects, representing the buttons to display in the reactions bar.

```
var textReactions = [  
  {  
    text: 'Inspiring',  
    ID: 'inspiring',  
  
    iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/Inspiring_Icon_Up.png',  
  
    iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/Inspiring_Icon_Down.png',  
    tooltip: 'This item is inspiring',  
    feedMessage: 'Inspiring!',  
    headerText: 'You think this post is inspiring,'  
  },  
  {  
    text: 'Innovative',  
    ID: 'innovative',  
  
    iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/Innovative_Icon_Up.png',  
  
    iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/Innovative_Icon_Down.png',  
    tooltip: 'This item is innovative',  
    feedMessage: 'This is innovative',  
    headerText: 'You think this post is innovative,'  
  },  
  {  
    text: 'Amazing',  
    ID: 'amazing',  
  
    iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/Amazing_Icon_Up.png',  
  
    iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/Amazing_Icon_Down.png',  
    tooltip: 'This item is amazing',  
    feedMessage: 'This is amazing!',  
    headerText: 'You think this post is amazing,'  
  }  
]
```

```
, {
  text: 'LOL',
  ID: 'lol',

  iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/LOL_Icon_Up.png',

  iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/LOL_Icon_Down.png',
  tooltip: 'This made me LOL',
  feedMessage: 'LOL!',
  headerText: 'This post made you LOL,'
}
, {
  text: 'Geeky',
  ID: 'geeky',

  iconImgUp: 'https://cdns.gigya.com/gs/i/reactions/icons/Geeky_Icon_Up.png',

  iconImgOver: 'https://cdns.gigya.com/gs/i/reactions/icons/Geeky_Icon_Down.png',
  tooltip: 'This item is geeky',
  disabledTooltip: 'You found this geeky',
  feedMessage: 'This is geeky!',
  headerText: 'You think this is geeky,'
```

```
}  
];
```

Each Reaction object has a unique ID (unique within the scope of this method call). You may define for each Reaction object text and/or an icon that will be presented on the corresponding reaction button. The Reaction object includes additional optional fields. Please refer to the [Reaction object](#) reference for the complete list of optional fields.

Refer to the [Buttons Design](#) section to learn how to fully customize and apply your own design to the buttons.

## Step 2: Construct a UserAction Object and fill it with data

The `UserAction` object represents a user action. We use the `UserAction` object constructor for this purpose:

```
var defaultUserAction = new gigya.socialize.UserAction();
```

After you have constructed the object, you can use the various `UserAction` [member methods](#) to set the object's properties. For example:

```
defaultUserAction.setTitle("This is my title");  
...
```

The `UserAction`'s object properties represent the various items of which a User Action may consist: Title, etc.

Please refer to the [Advanced Sharing](#) guide to learn more about the `UserAction` object and how to implement the full richness of newsfeed reports, including adding various media items (Images/ a Flash file/ Mp3 file/ a Video).

## Step 3: Define the Reactions add-on's parameters

```
var reactionParams = {  
    barID: 'today-story', // Identifier of the content to which this  
    reaction refers  
    containerID: 'reactionsDiv', // Reactions add-on DIV Container  
    reactions: textReactions, // The reaction array from Step 1  
    userAction: defaultUserAction, // The UserAction object from Step  
    2  
    bodyText: 'Share it with your friends:', // optional - text that  
    appears in the Share popup  
    showCounts: 'top', // optional - displays the counters on top of  
    the buttons  
    cid:''  
};
```

The Reactions add-on has four required parameters:

- *containerID* - the identifier of the DIV element, where the Reactions add-on should be embedded.
- *reactions* - the reaction array that we have created in Step 1.
- *userAction* - the `userAction` object that we have created in Step 2.
- *barID* - identifier of the content to which this reaction refers.

In addition there is a wide list of optional parameters. In our example we have used the *'bodyText'* optional parameter. This parameter defines the second line of text that appears on the mini Share pop-up dialog (see screenshot above). Please refer to the [method's parameters](#) table for the entire list of optional parameters.

## Step 4: Invoke the Reactions add-on

Call the `socialize.showReactionsBarUI` API method:

```
gigya.socialize.showReactionsBarUI(reactionParams);
```

Like every Gigya API method, the `socialize.showReactionsBarUI` method receives one parameter - `parameters` object, which is the object we have defined in Step 3 (learn more about Gigya's basics [here](#)).

## The Published Reactions on Facebook

### UserAction per Reaction Bar

In the example above there is one `UserAction` that is used for any reaction in the reaction bar:

```
// Define UserAction object and fill it with data
var defaultUserAction = new gigya.socialize.UserAction();
defaultUserAction.setTitle("This is the post title");
defaultUserAction.setLinkBack("http://www.gigya.com");
```

The `defaultUserAction` that is defined here is used as the `UserAction` in the reaction bar's parameters.

This following screenshot shows the user reaction ("Amazing"), which we have constructed in the above example, as it appears when published on Facebook:



The text "This is amazing" is the `feedMessage` that is defined in the button representing the reaction "Amazing".

### UserAction per Reaction Button

You can also construct a different `UserAction` for each reaction button, which will override the `UserAction` in the reactions bar parameters:

```
// UserAction to use when reaction is "love it"
var UA_love_it = new gigya.socialize.UserAction();
UA_love_it.setTitle("I love it");
UA_love_it.setLinkBack("http://www.gigya.com");
```

In this case the user reaction as it appears in Facebook will be as follows:



Tanya Rosing

I love it

 Like · Comment · A few seconds ago via Gigya Socialize · 

## Adding an Image to Published Reaction

When you construct a different UserAction for each reaction button you can also add the image of the reaction by defining an image object and then using it in the UserAction:

```
// The image to show in share when reaction is "love it"
var image_love_it = {

src: 'https://cdns.gigya.com/gs/i/reactions/icons/LoveIt_Icon_Up.png',

href: 'https://cdns.gigya.com/gs/i/reactions/icons/LoveIt_Icon_Up.png',
  type: 'image'
};

// UserAction to use when reaction is "love it"
var UA_love_it = new gigya.socialize.UserAction();
UA_love_it.setTitle("I love it");
UA_love_it.setLinkBack("http://www.gigya.com");
UA_love_it.addMediaItem(image_love_it);
```

In this case the user reaction as it appears in Facebook will include the image:



Tanya Rosing

 I love it

 Like · Comment · 4 seconds ago via Gigya Socialize · 

**Note:** Even though the UserAction in this example is defined separately for each button, you must still define a UserAction in the parameters of the reaction bar because it is a **required** parameter. This parameter is overridden by the UserAction that each button has.

You can see a working example where each button has its own UserAction in the [Reactions - Voting](#) example.

## Allow Clicking Only One Reaction

In certain cases only one reaction button should be allowed to be clicked on the same reaction bar, for example when you want to vote either for an item or against an item. To implement this set the **multipleReactions** parameter to 'false':

```
var iconReactionsParams = {
  multipleReactions: 'false',
  barID: 'myID',
  containerID: 'iconReactionsDiv',
  reactions: iconReactions,
  userAction: UA
};
```

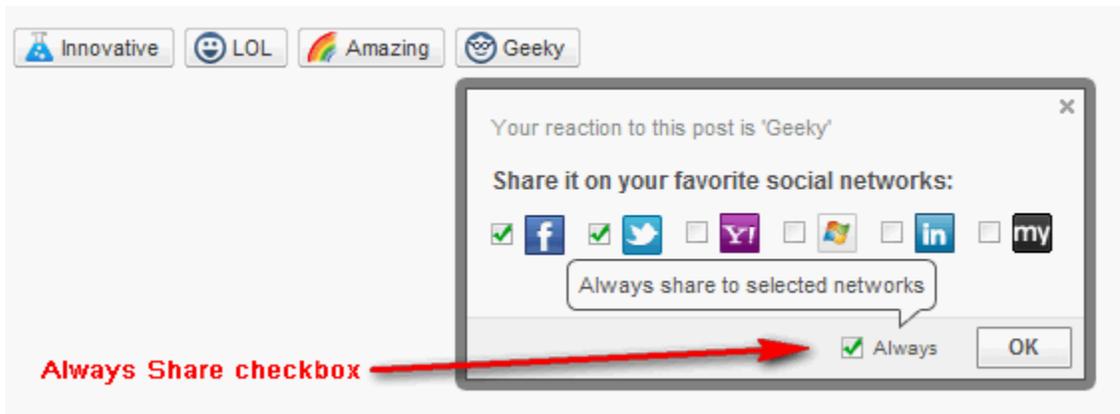
Another example where only one reaction button should be allowed to be clicked on the same reaction bar is a poll. You can see a working example where only one reaction can be clicked in the [Reactions - Voting](#) example and the [Reactions - Poll](#) example.

## A Working Example

In the [Reactions add-on demo](#) you will find a complete working example of a web-page/widget that uses the Reactions add-on. You may view the code, run it and view the outcome. There are two additional examples using the reactions add-on, one which allows the user to vote either against or for an item in the [Reactions - Voting](#) example, the second allows the user to participate in a poll in the [Reactions - Poll](#) example.

## Automatic Sharing

Gigya offers an option to activate an automatic share per user and per predefined type of activity in your site. Automatic share means that each time the user does the same type of activity in your site (i.e. clicks on a certain Reaction button), his action will be automatically shared on pre-selected social networks, without prompting the mini Share add-on. Obviously, automatic share activation must receive the user's permission:



This feature is enabled and customized through the use of `socialize.showReactionsBarUI`'s auto share related parameters. This feature is identical to the parallel feature in the Share add-on. To implement the automatic sharing feature in the Reactions add-on, follow the [Automatic Sharing \(in the Share add-on\)](#) guide, substituting the Share add-on (`socialize.showShareUI` method) with the Reaction add-on (`socialize.showReactionsBarUI` method). The guide includes running code examples and comprehensive explanations.

## Using the add-on to Initiate Site Login

Please read the [Using add-ons to Initiate Site Login](#) guide to learn how to integrate the add-on with the [Social Login](#) process and leverage the add-on to acquire new site users.