jQueryUI provides droppable method to make any DOM element droppable at a specified target.

### Syntax

The **droppable** method can be used in two forms:

- **$**(selector, context)**.droppable**(options)** Method
- **$**(selector, context)**.droppable**("action", params)** Method

#### $**(selector, context)**.droppable**(options)** Method

The **droppable options** method declares that an HTML element can be used as an element in which you can drop other elements. The **options** parameter is an object that specifies the behavior of the elements involved.

### Syntax

```javascript
$(selector, context).droppable(options);
```

You can provide one or more options at a time using Javascript object. If there are more than one options to be provided then you will separate them using a comma as follows:

```javascript
$(selector, context).droppable({option1: value1, option2: value2...... });
```

Following table lists the different **options** that can be used with this method:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>accept</strong></td>
<td>This option is used when you need to control which draggable elements are to be accepted for dropping. By default its value is *.</td>
</tr>
<tr>
<td><strong>activeClass</strong></td>
<td>This option is a String representing one or more CSS classes to be added to the droppable element when an accepted element (one of those indicated in options.accept) is being dragged. By default its value is <strong>false</strong>.</td>
</tr>
<tr>
<td><strong>addClasses</strong></td>
<td>This option when set to <strong>false</strong> will prevent the ui-droppable class from being added to the droppable elements. By default its value is <strong>true</strong>.</td>
</tr>
<tr>
<td><strong>disabled</strong></td>
<td>This option when set to <strong>true</strong> disables the droppable. By default its value is <strong>false</strong>.</td>
</tr>
<tr>
<td><strong>greedy</strong></td>
<td>This option is used when you need to control which draggable elements are to be accepted for dropping on nested droppables. By default its value is <strong>false</strong>. If this option is set to <strong>true</strong>, any parent droppables will not receive the element.</td>
</tr>
<tr>
<td><strong>hoverClass</strong></td>
<td>This option is <strong>String</strong> representing one or more CSS classes to be added to the element of droppable when an accepted element (an element indicated in options.accept) moves into it. By default its value is <strong>false</strong>.</td>
</tr>
</tbody>
</table>
scope
This option is used to restrict the droppable action of draggable elements only to items that have the same options.scope definedindraggable(options). By default its value is "default".

tolerance
This option is a String that specifies which mode to use for testing whether a draggable is hovering over a droppable. By default its value is "intersect".

Following section will show you few working examples of drop functionality.

Default Functionality

The following example demonstrates a simple example of droppable functionality, passing no parameters to the droppable method.

```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI Droppable - Default functionality</title>
    <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
    <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
    <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <style>
    #draggable-1 {
        width: 100px; height: 50px; padding: 0.5em; float: left;
        margin: 22px 5px 10px 0;
    }
    #droppable-1 {
        width: 120px; height: 90px; padding: 0.5em; float: left;
        margin: 10px;
    }
    </style>
    <script>
        $(function() {
            $('#draggable-1').draggable();
            $('#droppable-1').droppable();
        });
    </script>
</head>
<body>
    <div>
        <p>Drag me to my target</p>
    </div>
    <div>
        <p>Drop here</p>
    </div>
</body>
</html>
```

Let's save above code in an HTML file dropexample.htm and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

Drag me to my target
Drop here

Use of addClass, disabled and tolerance

The following example demonstrates the usage of three options a addClass b disabled and c tolerance in the drop function of JqueryUI.
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Droppable - Default functionality</title>
<link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
<script src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<style>
#draggable-2 {
  width: 100px; height: 50px; padding: 0.5em;
  margin: 0px 5px 10px 0;
}
#droppable-2, #droppable-3, #droppable-4, #droppable-5 {
  width: 120px; height: 90px; padding: 0.5em; float: left;
  margin: 10px;
}
</style>
<script>
$( function() {
  $( "#draggable-2" ).draggable();
  $( "#droppable-2" ).droppable({
    drop: function( event, ui ) {
      $( this ).
        .addClass( "ui-state-highlight" )
        .find( "p" )
        .html( "Dropped!" );
    }
  });
  $( "#droppable-3" ).droppable({
    disabled : "true",
    drop: function( event, ui ) {
      $( this ).
        .addClass( "ui-state-highlight" )
        .find( "p" )
        .html( "Dropped!" );
    }
  });
  $( "#droppable-4" ).droppable({
    tolerance: 'touch',
    drop: function( event, ui ) {
      $( this ).
        .addClass( "ui-state-highlight" )
        .find( "p" )
        .html( "Dropped with a touch!" );
    }
  });
  $( "#droppable-5" ).droppable({
    tolerance: 'fit',
    drop: function( event, ui ) {
      $( this ).
        .addClass( "ui-state-highlight" )
        .find( "p" )
        .html( "Dropped only when fully fit on the me!" );
    }
  });
});
</script>
</head>
<body>
  <div>
    <p>Drag me to my target</p>
  </div>
  <div>
    <p>Drop here</p>
  </div>
</body>
Let's save above code in an HTML file dropexample.htm and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

Drag me to my target
Drop here
I'm disabled, you can't drop here!
Tolerance Touch!
Tolerance Fit!

Now drop the element on the "Tolerance Touch!" box justtouchtheedgeofthisbox and see the changes of target element. Now to drop the element on "Tolerance Fit!" target, the draggable element has to fully overlap the target element i.e "Tolerance Fit!" target.

Choose elements to be dropped

The following example demonstrates the use of option accept and scope option in the drag function of JqueryUI.
$( "#india" ).droppable({
    accept: "#indiapeople",
    drop: function( event, ui ) {
        $( this )
            .addClass( "ui-state-highlight" )
            .find( "p" )
            .html( "Dropped!");
    }
});

$( "#javatutorial" ).draggable({scope : "java"});
$( "#springtutorial" ).draggable({scope : "spring"});
$( "#java" ).droppable({
    scope: "java",
    drop: function( event, ui ) {
        $( this )
            .addClass( "ui-state-highlight" )
            .find( "p" )
            .html( "Dropped!");
    }
});

$( "#spring" ).droppable({
    scope: "spring",
    drop: function( event, ui ) {
        $( this )
            .addClass( "ui-state-highlight" )
            .find( "p" )
            .html( "Dropped!");
    }
});
</script>
</head>
<body>
<div>
    <div>
        <p>People to be dropped to Japan</p>
    </div>
    <div>
        <p>People to be dropped to India</p>
    </div>
    <div>
        <p>Japan</p>
    </div>
    <div>
        <p>India</p>
    </div>
    <hr/>
    <div>
        <p>People who want to learn Java</p>
    </div>
    <div>
        <p>People who want to learn Spring</p>
    </div>
    <div>
        <p>Java</p>
    </div>
    <div>
        <p>Spring</p>
    </div>
</div>
Let's save above code in an HTML file dropexample.htm and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the output:

People to be dropped to Japan
People to be dropped to India
Japan
India

People who want to learn Java
People who want to learn Spring
Java
Spring

Here you can see that you can drop element "People from Japan" on only "Japan" target and element "People from India" on target India. Similary the scope for "People who want to learn Java" is set to target "Java" and "People who want to learn Spring" is set to "Spring target".

Managing appearance

Following example demonstrates the use of options activeClass and hoverClass of JqueryUI class, which help us manage appearance.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>jQuery UI Droppable - Default functionality</title>
    <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
    <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
    <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <style type="text/css">
      #draggable-3 {
        width: 100px; height: 50px; padding: 0.5em; float: left;
        margin: 21px 5px 10px 0;
      }
      #droppable-6 {
        width: 120px; height: 90px; padding: 0.5em; float: left;
        margin: 10px;
      }
      .active {
        border-color: blue;
        background: grey;
      }
      .hover {
        border-color: red;
        background: green;
      }
    </style>
  </head>
  <script>
    $(function() {
      $('.draggable-3').draggable();
      $('.draggable-6').droppable({
        activeClass: "active",
        hoverClass: "hover",
        drop: function( event, ui ) {
          // Your drop functionality
        }
      });
    });
  </script>
</html>
```
Let's save above code in an HTML file dropexample.htm and open it in a standard browser which supports javascript, you must see the following output:

Drag me to my target
Drop here

You can notice that on dragging or hovering over the target of "Drag me to my target" element, changes the color of target element "Drop here".

$(selector, context).droppable ("action", params) Method

The droppable "action", params method can perform an action on droppable elements, such as preventing droppable functionality. The action is specified as a string in the first argument e.g., "disable" to prevent the drop. Check out the actions that can be passed, in the following table.

Syntax

$(selector, context).droppable ("action", params);

Following table lists the different actions that can be used with this method:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destroy</td>
<td>This action destroys the droppable functionality of an element completely. The elements return to their pre-init state.</td>
</tr>
<tr>
<td>disable</td>
<td>This action disables the droppable operation. The elements are no longer droppable elements. This method does not accept any arguments.</td>
</tr>
<tr>
<td>enable</td>
<td>This action reactive the droppable operation. The elements can again receive a droppable element. This method does not accept any arguments.</td>
</tr>
<tr>
<td>option</td>
<td>This action gets an object containing key/value pairs representing the current droppable options hash. This method does not accept</td>
</tr>
</tbody>
</table>
option optionName

This action gets the value of currently associated droppable element with the specified optionName. This takes a String value as argument.

option optionName, value

This action sets the value of the droppable option associated with the specified optionName.

option options

This action is sets one or more options for the droppable. The argument options is a map of option-value pairs to be set.

widget

This action returns a jQuery object containing the droppable element. This method does not accept any arguments.

Example

Now let us see an example using the actions from the above table. The following example demonstrates the use of destroy method.

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Droppable - Default functionality</title>
  <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <style>
    .draggable-4 {
      width: 90px; height: 50px; padding: 0.5em; float: left;
      margin: 0px 5px 10px 0;
      border: 1px solid red;
      background-color:#B9CD6D;
    }
    .droppable-7 {
      width: 100px; height: 90px;padding: 0.5em; float: left;
      margin: 10px;
      border: 1px solid black;
      background-color:#A39480;
    }
    .droppable.active {
      background-color: red;
    }
  </style>
  <script>
    $(function() {
      $('.draggable-4').draggable({ revert: true });
      $('.droppable-7').droppable({
        hoverClass: 'active',
        drop: function(e, ui) {
          $(this).html(ui.draggable.remove().html());
          $(this).droppable('destroy');
          $( this )
            .addClass( "ui-state-highlight" )
            .find( "p" )
            .html( "i'm destroyed!" );
        }
      });
    });
</script>
</head>
<body>
</body>
</html>
```
Let's save above code in an HTML file `dropexample.htm` and open it in a standard browser which supports javascript, you must see the following output:

drag 1
drag 2
drag 3
drop here
drop here
drop here

If you drop "drag1" on any of the elements named "drop here", you will notice that this element gets dropped and this action destroys the droppable functionality of an element completely. You cannot drop "drag2" and "drag3" on this element again.

### Event Management on droppable elements

In addition to the droppable `options` method we saw in the previous sections, JqueryUI provides event methods which gets triggered for a particular event. These event methods are listed below:

<table>
<thead>
<tr>
<th>Event Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>activate</strong> event, ui</td>
<td>This event is triggered when the accepted draggable element starts dragging. This can be useful if you want to make the droppable “light up” when it can be dropped on.</td>
</tr>
<tr>
<td><strong>create</strong> event, ui</td>
<td>This event is triggered when a droppable element is created. Where <code>event</code> is of type <code>Event</code>, and <code>ui</code> is of type <code>Object</code>.</td>
</tr>
<tr>
<td><strong>deactivate</strong> event, ui</td>
<td>This event is triggered when an accepted draggable stops dragging. Where <code>event</code> is of type <code>Event</code>, and <code>ui</code> is of type <code>Object</code>.</td>
</tr>
<tr>
<td><strong>drop</strong> event, ui</td>
<td>This action is triggered when an element is dropped on the droppable. This is based on the <code>tolerance</code> option. Where <code>event</code> is of type <code>Event</code>, and <code>ui</code> is of type <code>Object</code>.</td>
</tr>
<tr>
<td><strong>out</strong> event, ui</td>
<td>This event is triggered when an accepted draggable element is</td>
</tr>
</tbody>
</table>
dragged out of the droppable. This is based on the tolerance option. Where event is of type Event, and ui is of type Object.

over event, ui

This event is triggered when an accepted draggable element is dragged over the droppable. This is based on the tolerance option. Where event is of type Event, and ui is of type Object.

Example

The following example demonstrates the event method usage during drop functionality. This example demonstrates the use of events drop, over and out.

```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI Droppable - Default functionality</title>
    <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
    <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
    <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <style>
        #draggable-5 {
            width: 100px; height: 50px; padding: 0.5em; float: left;
            margin: 22px 5px 10px 0;
        }
        #droppable-8 {
            width: 120px; height: 90px; padding: 0.5em; float: left;
            margin: 10px;
        }
    </style>
    <script>
        $(function() {
            $('#draggable-5').draggable();
            $('#droppable-8').droppable({
                drop: function (event, ui) {
                    $(this).
                        .addClass( "ui-state-highlight" )
                        .find( "p" )
                        .html( "Dropped!" );
                },
                over: function (event, ui) {
                    $(this).
                        .addClass( "ui-state-highlight" )
                        .find( "p" )
                        .html( "moving in!" );
                },
                out: function (event, ui) {
                    $(this).
                        .addClass( "ui-state-highlight" )
                        .find( "p" )
                        .html( "moving out!" );
                }
            });
        });
    </script>
</head>
<body>
    <div>
        <p>Drag me to my target</p>
    </div>
</body>
</html>
```
Let's save above code in an HTML file `dropexample.htm` and open it in a standard browser which supports javascript, you must see the following output:

Drag me to my target

Drop here

Here you will notice how the message in the droppable element changes as you drag the element.