

SERVICE CLOUD LEARNING

EMAIL - to - Case for customer

dhakedpiyush9291@n7wo1xa33upbjcdzaniunionr8m183xwoz4zsotujrkiditjmj.5i-3mhvheaw.ap26.case.salesforce.com

The screenshot shows the 'Email-to-Case' setup page in the Salesforce Setup interface. It includes sections for:

- Settings:** Includes checkboxes for 'Enable Email-to-Case' (checked), 'Notify Case Owners on New Emails' (unchecked), 'Enable HTML Email' (checked), 'Set Case Source to Email' (checked), and 'Save Email-to-Case Attachments as Salesforce Files' (unchecked).
- Send Emails from Cases:** Includes a checkbox for 'Place User Signatures before Email Threads' (unchecked).
- On-Demand Service:** Includes a checkbox for 'Enable On-Demand Service' (checked). Below it, it shows 'Over Email Rate Limit Action' set to 'Bounce message' and 'Unauthorized Sender Action' set to 'Discard message'.
- Routing Addresses:** A table listing a single routing address:

Action	Source	Routing Name	Case Owner	Email Address	Verification	Email Services Address
Edit Del	Email2Case	Customer Support Email version	asf asfsa	dhakedpiyush9291@gmail.com	Verified	dhakedpiyush9291@n7wo1xa33upbjcdzaniunionr8m183xwoz4zsotujrkiditjmj.5i-3mhvheaw.ap26.case.salesforce.com

- 1) Escalation Rules - If a case is not addressed within a time limit it is assigned to other user or queue

The screenshot shows the 'Escalation Rules' setup page in the Salesforce Setup interface. It includes:

- Case Escalation:** A rule entry with the following details:

Rule Name	Case Escalation
Order	1
Rule Criteria	Case: Subject CONTAINS hardware
Business Hours Settings	Use business hours specified on the case
How escalation times are set	When case is created
- Escalation Actions:** A table showing an action:

Action	Escalate At	Assign To	Email	Notify	Template
Edit Del	30 Minutes	Service Case	<input type="checkbox"/>		

- 2) Gathering cases through - Email-to-Case - We have to enable "Enable Email-to-Case" and On service demand after that we get that email on which user will send Email for Query.

The screenshot shows the Salesforce Setup interface under the 'Email-to-Case' section. The page title is 'Email-to-Case Routing Address'. It displays the following details:

- Email Address Detail:** Edit, Delete, Clone buttons.
- Routing Information:** Source: Email2Case.
- Notes:** A note states: "Salesforce.com automatically created the following email service address for you: dhakedpiyush9291@n7wo1xa33upbjcdzaniunonr8m183xwoz4zsotujkidljm|5i-3mhvheaw.ap26.case.salesforce.com. Configure your email system to forward messages received at dhakedpiyush9291@gmail.com to this email services address."
- Routing Name:** Customer Support Email version
- Email Address:** dhakedpiyush9291@gmail.com [Verified]
- Email Services Address:** dhakedpiyush9291@n7wo1xa33upbjcdzaniunonr8m183xwoz4zsotujkidljm|5i-3mhvheaw.ap26.case.salesforce.com
- Created By:** Piyush Dhaked, 9/14/2022, 2:53 PM
- Modified By:** Piyush Dhaked, 9/14/2022, 2:54 PM
- Email Settings:** Save Email Headers (checkbox), Accept Email From (checkbox).
- Task Settings:** Create Task from Email (checkbox), Task Status: Not Started.
- Case Settings:** Case Owner: asf-asfsa, Case Priority: High, Case Origin: Email.

- 3) Gathering cases through - Web-to-Case - We have to enable web to case in Home -> Quick Find. And after that Web-to-case HTML generator will give us code. That we have to add in website.

Web-to-Case Settings

Use a simple web form or a self-service customer community to make it easy for customers to submit cases directly to your customer support group. For detailed information on setting up Web-to-Case or Self-Service Community templates, see the Salesforce help.

Basic Settings

- Enable Web-to-Case: ⓘ
- Require reCAPTCHA Verification: ⓘ
- Default Case Origin: Web ⓘ

Auto-Response Email Settings

- Default Response Template: Web-to-Case Settings ~ Salesforce - Developer Edition ⓘ
- Hide Record Information: ⓘ
- Email Signature: ⓘ

Save Cancel

 **SETUP**
Web-to-Case HTML Generator

Capturing Case Information from Your Website Help for this Page 

Using pre-existing pages on your company's website, you can capture contact and case information from users and automatically create new cases in salesforce.com, enabling you to respond in real-time to customer requests.

Capture Cases

Select the fields to include:

Available Fields	Selected
Company	Contact Name
Type	Email
Status	Phone
Case Reason	Subject
Priority	Description
Case Currency	
Case Record Type	
Engineering Req Number	
SLA Violation	

Note: Would you like to add custom fields that you do not see listed under Available Fields? You can set up custom lead fields to gather additional information from your website. [Tell me more](#).

Visible in Self-Service Portal

Enter the URL that the user will be returned to:
URL:

Include reCAPTCHA in HTML 

Generate **Cancel**

Web-to-Case Form

Contact Name  First Name

E-Mail  E-Mail Address

Contact No.  Phone

Subject  Subject

Description 

SUBMIT 

4) Autoresponse Rules - We can send a notification Email for anything that happens with case i.e If a case gets escalated Send Email to the case owner.

The screenshot shows the 'Case Auto-Response Rules' setup page. At the top, there's a header with a gear icon labeled 'SETUP' and the page title 'Case Auto-Response Rules'. Below the header, a sub-header reads 'Case Auto-Response Rule demo 1'. A note says 'Add rule entries that specify the criteria and email template to use to respond to cases. You can reorder rule entries on this page after you create them.' On the left, a 'Rule Detail' section shows 'Rule Name: demo 1', 'Created By: Piyush Dhaked, 9/12/2022, 6:41 PM', 'Active: checked', and 'Modified By: Piyush Dhaked, 9/14/2022, 10:58 PM'. On the right, a 'Rule Entries' table lists one entry:

Action	Order	Criteria	Sent From (Email)	Template
Edit Del	1	Case: Escalated EQUALS True	Aetherenus (piyushfeast9291@gmail.com)	Support_Escalated_Case_Notification

5) Entitlements - With Entitlements support agents can determine whether a customer is eligible for support or not

6) Entitlement Processes - With these you can design the timeline that includes all steps that you want your support team must complete to resolve support records like cases.

7) Entitlement Templates - With this we don't have to create Entitlement every-time. We can make a template and assign that.

8) Milestones - These are metrics that represent service levels to provide to each of your customers.

They represent time-dependent steps in your support Process like first response or case resolution time

9) Knowledge - Whenever a service team closes a case we have a place known as knowledge where they can share this case as other members also encountered same cases. So Whenever a similar case arrives, agent can refer to that knowledge article.

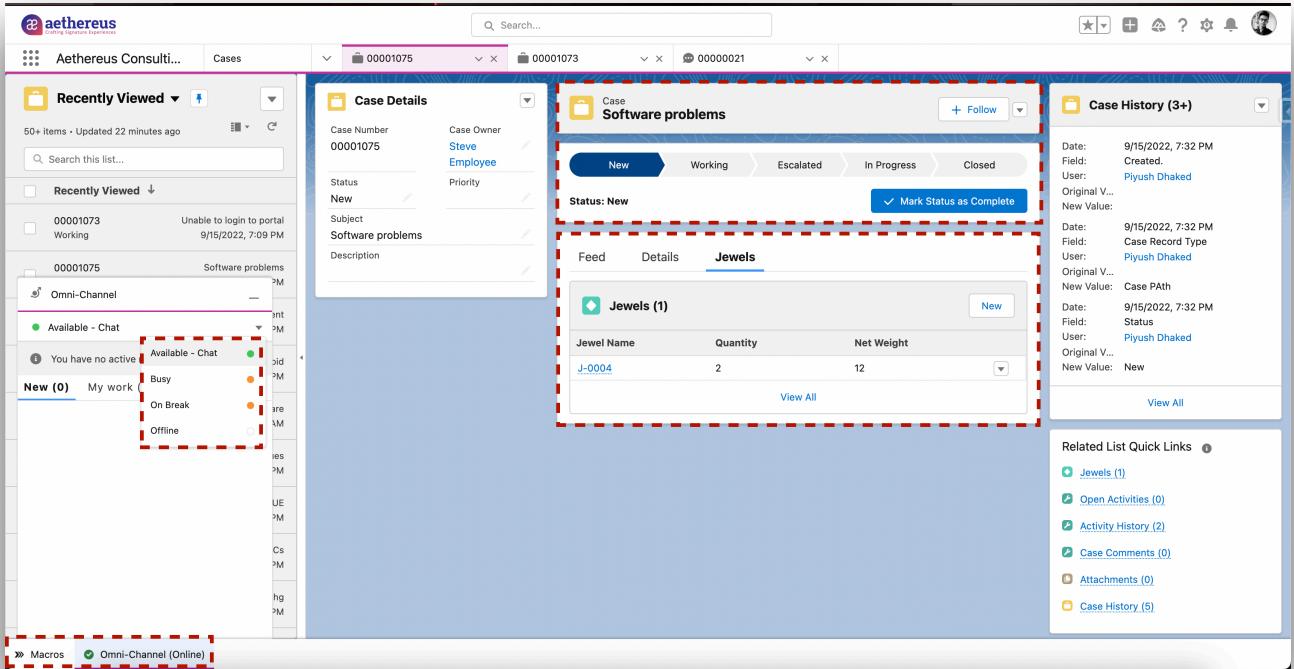
10) There is a Queue - Customer Support Team, 2 users - "Steve Employee" and "ASF asfsa", and a System Admin "Piyush Dhaked".

Assignment rule says - If case : subject contains hardware assign that case to customer support team queue.

There is Email to case scenario - And in Email Subject is hardware - In this case Case will be assigned to user "ASF asfsa" because it is mentioned in the Email-to-case rule.

But in Web-to-case there is no Specific assignment rule so we follow assignment rule entries.

Final Product



Web-to-case Code

```

<!DOCTYPE html>
<html>

<head>
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"
        integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
        crossorigin="anonymous">

  <!-- Optional theme -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap-theme.min.css"
        integrity="sha384-BVHyNvH2XZT28aJNxWGLyUx43JwzGj6uJ/EEq50Ik/Sp"
        crossorigin="anonymous">

  <!-- Latest compiled and minified JavaScript -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"
        integrity="sha384-"
        Tc5IQib027qvjSMfHjOMaLkfuWVxZxUPnCJA7I2mCWNIpG9mGCD8wGNlcPD7Txa"

```

```
crossorigin="anonymous"></script>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width">
<title>replit</title>
<style>
  #success_message {
    display: none;
  }
</style>
</head>

<body>
  <div class="container">

    <form class="well form-horizontal" action="https://webto.salesforce.com/servlet/servlet.WebToCase?encoding=UTF-8"
      method="post" id="contact_form">
      <input type=hidden name="orgid" value="00D5i000003MHvH">
      <input type=hidden name="retURL" value="http://www.aethereus.com">

      <fieldset>

        <legend>
          <center>
            <h2><b>Web-to-Case Form</b></h2>
          </center>
        </legend><br>

        <div class="form-group">
          <label class="col-md-4 control-label">Contact Name</label>
          <div class="col-md-4 inputGroupContainer">
            <div class="input-group">
              <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>
              <input placeholder="First Name" maxlength="80" name="name" size="20" class="form-control" type="text">
            </div>
          </div>
        </div>

        <div class="form-group">
          <label class="col-md-4 control-label">E-Mail</label>
```

```
<div class="col-md-4 inputGroupContainer">
  <div class="input-group">
    <span class="input-group-addon"><i class="glyphicon glyphicon-envelope"></i></span>
    <input maxlength="80" name="email" size="20" placeholder="E-Mail Address" class="form-control" type="text">
  </div>
</div>
</div>
```

```
<div class="form-group">
  <label class="col-md-4 control-label">Contact No.</label>
  <div class="col-md-4 inputGroupContainer">
    <div class="input-group">
      <span class="input-group-addon"><i class="glyphicon glyphicon-earphone"></i></span>
      <input maxlength="40" name="phone" size="20" placeholder="Phone" class="form-control" type="text">
    </div>
  </div>
</div>
```

```
<div class="form-group">
  <label class="col-md-4 control-label">Subject</label>
  <div class="col-md-4 inputGroupContainer">
    <div class="input-group">
      <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>
      <input maxlength="80" name="subject" size="20" placeholder="Subject" class="form-control" type="text">
    </div>
  </div>
</div>
```

```
<div class="form-group">
  <label class="col-md-4 control-label">Description</label>
  <div class="col-md-4 inputGroupContainer">
```



```
    notEmpty: {
      message: 'Please enter your First Name'
    }
  },
  last_name: {
    validators: {
      stringLength: {
        min: 2,
      },
      notEmpty: {
        message: 'Please enter your Last Name'
      }
    }
  },
  user_name: {
    validators: {
      stringLength: {
        min: 8,
      },
      notEmpty: {
        message: 'Please enter your Username'
      }
    }
  },
  user_password: {
    validators: {
      stringLength: {
        min: 8,
      },
      notEmpty: {
        message: 'Please enter your Password'
      }
    }
  },
  confirm_password: {
    validators: {
      stringLength: {
        min: 8,
      },
      notEmpty: {
        message: 'Please confirm your Password'
      }
    }
  },
  email: {
    validators: {
      notEmpty: {
```

```

        message: 'Please enter your Email Address'
    },
    emailAddress: {
        message: 'Please enter a valid Email Address'
    }
},
},
contact_no: {
    validators: {
        stringLength: {
            min: 12,
            max: 12,
            notEmpty: {
                message: 'Please enter your Contact No.'
            }
        }
    }
},
department: {
    validators: {
        notEmpty: {
            message: 'Please select your Department/Office'
        }
    }
},
}
})
.on('success.form.bv', function (e) {
    $('#success_message').slideDown({opacity: "show"}, "slow") // Do something ...
    $('#contact_form').data('bootstrapValidator').resetForm();

    // Prevent form submission
    e.preventDefault();

    // Get the form instance
    var $form = $(e.target);

    // Get the BootstrapValidator instance
    var bv = $form.data('bootstrapValidator');

    // Use Ajax to submit form data
    $.post($form.attr('action'), $form.serialize(), function (result) {
        console.log(result);
    }, 'json');
});
});

</script>

```

</html>