

Table of Contents

- [Overview](#)
- [Customer Facing Web Pages](#)
 - [index.html](#)
 - [acdwait.html](#)
 - [invalidsession.html](#)
 - [ininvid/injector.js](#)
 - [ininvid/style.part.html](#)
 - [ininvid/content.part.html](#)

Overview

This page will document the standard customization points within the integration.

Customer Facing Web Pages

This section will document the web pages that have been provided with the integration. These pages are intended to be modified or replaced for every deployment. This is necessary due to the highly customer-specific branding nature of web pages. The files described in this section can be found in the repo at [/src/DropInCode/](#).

index.html

This is a blank page illustrating a page that offers a video chat to the customer. This page has no actual content in the example; it only contains the minimum code necessary to inject the video chat code and is all present within the `<head>` element.

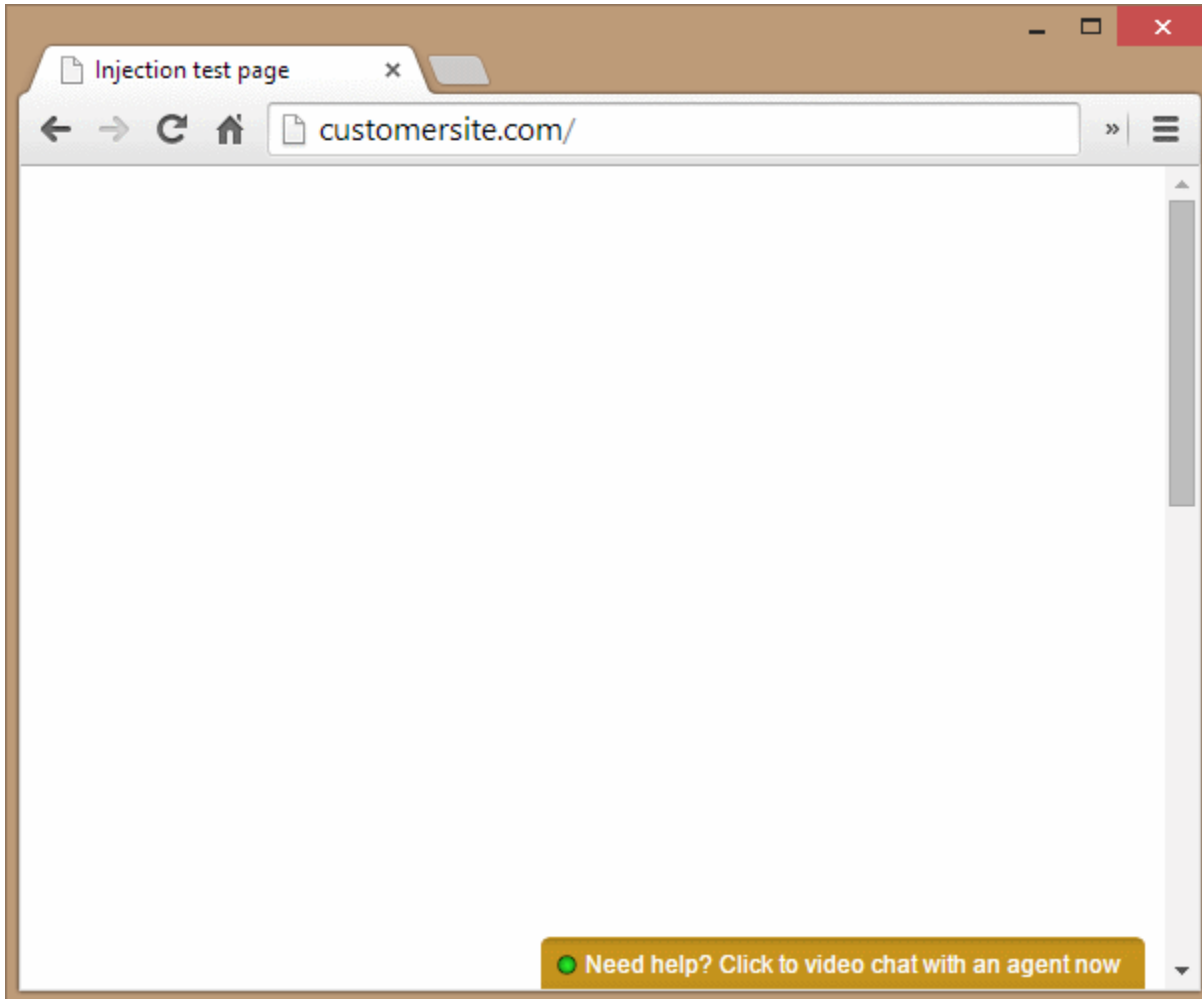
```
<script type="text/javascript" src="scripts/jquery-2.1.1.js"></script>
<script type="text/javascript" src="scripts/jquery.cookie-1.4.1.min.js"></script>
<script type="text/javascript" src="ininvid/injector.js"></script>
<script type="text/javascript">
  var ininvid_serverRoot = 'http://integrationserver.com:8000';
  var ininvid_displayName = 'Customer Name';
  var ininvid_reasons = {
    reasons: [
      { text:'Website Help', value:'Vidyo'},
      { text:'Customer Service', value:'support'},
      { text:'Sales Question', value:'sales'}
    ]
  };
</script>
```

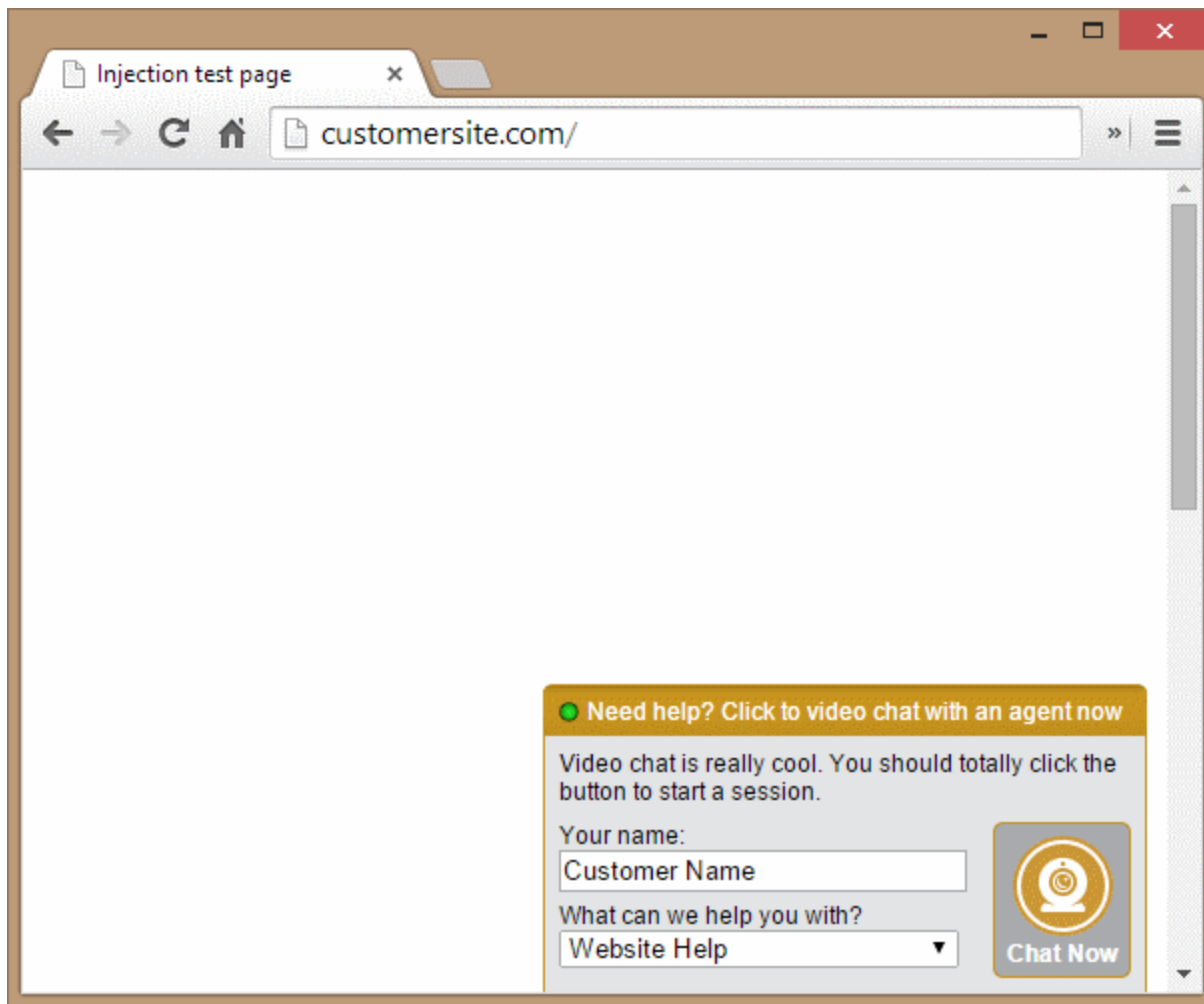
- Line 1: References the [jQuery library](#). This is not necessary if your site already uses jQuery.
- Line 2: References the [jQuery cookie extension library](#). This is not necessary if your site already uses the jQuery cookie extension library
- Line 3: References the injector JavaScript code. All of the code that gets injected into the page is initiated from this file.
- The rest of the script sets up a few configuration parameters:
 - `ininvid_serverRoot` - The base URL to the integration service. This is the URL

configured for the CicServiceEndpointUri parameter in the [Integration Service configuration](#).

- inivid_displayName - This is the display name for the guest. Typically, this will not be set statically, but will be set either by the user providing a name (as is included in the injected code) or by the page setting the variable behind the scenes based on metadata about the logged in user.
- inivid_reasons - This is a JSON document containing a collection of display names and CIC workgroup names to be presented in the dropdown in the injected code. This allows the user to select their destination queue.

Screenshots of the injector test page showing the injected widget in its collapsed state as well as its expanded state.





acdwait.html

This page is the landing page where a customer waits while the interaction is queuing for an agent. This page is intended as an example of what could be done with a wait page; each customer will have their own requirements for content delivery while the customer is waiting.

There are two key parts to this page:

Cookie Check

The injector code creates a cookie that stores the video conversation information prior to redirecting to the ACD wait page. The first operation in the ACD wait page is to check for this cookie. If the cookie does not exist, the user is redirected to the invalid session page.

checkConversationStatus() function

This method is invoked every 1000ms to check the status of the conversation. If the response comes back indicating that an agent has joined (data.userOwner has a value), the website user is redirected to the VidyoWeb URL. Otherwise, the loop continues.

invalidsession.html

This is a simple page that the user is redirected to if the session is invalid. Similar to the ACD wait page, this is just an example of how such a page might function and should be implemented per the customer's design requirements.

ininvid/injector.js

This is where the proverbial rubber meets the road. The injector code adds a widget to the page that offers a video chat to the customer. This file may be used as-is if the technical manner in which a video chat is initiated is not being customized; this file is the JavaScript code only and does not contain design elements (though it references UI elements by ID). This script performs the following operations:

- Inject the stylesheet: *ininvid/style.part.html*
- Inject the HTML: *ininvid/content.part.html*
- Populate the "reasons list" with the values specified in the JavaScript var *ininvidreasons_*
- Register for click events on the widget's header using jQuery's *.click* function. This expands/contracts the widget when the header is clicked.
- Register for hover events on the widget's header using jQuery's *.hover* function. This "peeks" the widget when it is collapsed and the user hovers over the header. The intent is to indicate to the user that they can interact with the widget.
- Register for click events on the widget's "chat now" button using jQuery's *.click* function. The click event here initiates the process to request a video conversation.
 - The request is made to the integration using jQuery's *.ajax* function. This invokes *POST /ininvid/v1/conversations* to create a new conversation. Upon receiving a response, the response data is stored in a cookie and the user is redirected to the ACD wait page.

ininvid/style.part.html

This is the stylesheet for the widget. This can be easily customized to change the colors, sizes, or general styling of the widget.

ininvid/content.part.html

This is the HTML markup for the widget. This can be customized or replaced to change the design of the widget. Be sure to update the stylesheet and the injector code if element IDs are changed.