# Lead API
# User Guide

**Contents**

# Read This First!

***PLEASE READ* – you'll need to know the following:**

- ***For live badges at the event, use:***

  **https://www.compusystems.com/servlet/LeadAPIRequest?type=submit**

  - We also provide a means of testing your token.

- You should have, or have access to, the ***technical knowledge to integrate with an API***.

- Your scanner should be configured to ***read QR codes only*** to prevent the accidental reading of other codes.

- It is recommended that you ***store the badge ID with each record***.

- Any assistance with the API is ***charged at $120/hr***.

- ***CSI does not provide assistance*** with your application.

- The ***token*** is valid ***90 days before*** the show floor opens and ***24 hours after*** the show floor closes.

- Please ensure that you support all versions of our QR code - *the original and the new one with the event number*:
  - ***{CSI Marker}****|{Badge Number}|{First Name}|{Last Name}|{Exhibitor Identifier}|{Badge Version}…*
    - *Original version…* **{CSI Marker}** = CSICSI
    - *CSI Marker with event id…* **{CSI Marker}** = CSI{event #} - *example:* CSI387
      - *These numbers will change from event to event - please refer to your confirmation email for your event number.*
    - *Digital Badge version*
      - *The data within the QR code is the same as the Printed Badge with the exception of an additional field which is for CompuSystems' use only.*

  - Also, be certain that even though the CSI Marker is typically a 3-digit number, you should be able to support any number of characters.

- *We STRONGLY recommend that you test your program by scanning all sample codes provided **here**.*

- What you see as an "**event ID**", our technical staff see as "**evt_uid**". *They're the same*.

# Preface

CSI's ***Lead API***, the latest installment in our ESG product line, enables our customers to "rent" the ability to capture leads by directly integrating to CompuSystems' backend.

The API ensures that the exhibitor uses the API on a show-by-show basis, thereby preventing "phishing" for leads.  It was designed from the ground up to be as simple as possible for you, the exhibitor, to integrate into your software.

This guide provides you with a full set of instructions that will guide the developer on how to integrate the API.

***Purpose:***

Document the technical details of using the Lead API.

***Requirements:***

Technical knowledge to integrate with an API.
Scanner configured to read QR codes with other codes turned off.
Ability to accept typed input of  the Badge Number is required to use the "Manual Entry" feature.

***Audience:***

Technical personnel.

***Conventions used in this guide:***

`Fixed font`
Used to indicate technical data.

`Fixed font with color`
Used to indicate web services requests and responses.

**Bold blue/green font**
Used for must-read information.

# Technical Info

## Outline

The following provides you with a bird's-eye view of the Lead API.

### Leads API Token

To obtain a token, please contact CompuSystems. You'll receive an **event ID** for your show as well as a separate token for your booth.

Here's an example of a token: `b0883f925af870786df58ffc4105d107`

### QR Code Data Format (PRINTED)

The data within the QR code has a fixed element and potentially a variable element (depending on the event – some events will not have a variable element). The fixed element is used for lead submissions and the variable is for CSI use and should be ignored.

The format uses a marker in the first field to identify that this is a QR code from CompuSystems, the second holds the badge number, the third has the first name, the fourth has the last name, and the fifth has an Exhibitor company identifier. The remaining fields, if any, are for CSI use only and should be ignored.

The data within this format is delimited by a pipe "|", and **the number of pipes can vary from show to show.**

*{CSI Marker}|{Badge Number}|{First Name}|{Last Name}|{Exhibitor Identifier}…*

***Note:*** *The {Exhibitor Identifier} above is for CompuSystems' use.*

*Example 1 (original):*
`CSICSI|123456|John|Smith|0002839`

*Example 2 (new):*
`CSI437|123456|Jane|Doe|0006831`
Looks good guys

**Please be aware that SPACES might be present within the registration data!!!**

```
CSICSI | 123456 | John | Smith | 0002839
CSI529 | 123456 | John | Smith | 0002839
```

## QR Code Data Format (DIGITAL)

Certain shows map employ Digital badges in addition or in place of Printed badges. Registrants with Digital Badges will access and display their Digital Badge via the Google Pay or Apple Wallet apps on their mobile devices. Digital badges are designed to work in the same manner as a Printed Badge.

The data within the QR code is the same as the Printed Badge with the exception of an additional field which is for CompuSystems' use only.

The data within this format is delimited by a pipe "|", and **the number of pipes can vary from show to show.**

***{CSI Marker}***|*{Badge Number}|{First Name}|{Last Name}|{Exhibitor Identifier}|{Badge Version}…*

***Note:*** *The {Exhibitor Identifier} and {Badge version} above is for CompuSystems' use.*

*Example 1 (Digital ):*
`CSI437|123456|John|Smith|0006831|123456:V:1619472985170`

*Example 2 (Printed):*
`CSI437|123456|Jane|Doe|0006831`

**Please be aware that SPACES might be present within the registration data!!!**

`CSI529 | 123456 | John | Smith | 123456:V:1619472985170`

## Manual Lead Entry

It is also possible to submit a Lead by manually typing in the badge number (`badge_id`). In order to use this feature, the application must be updated to support the typed input of the badge number which is printed on the badge itself by the user.

Depending on the show, the badge number that is printed on the badge may also include an additional checksum digit when compared to the badge number contained in the QR code. This is done to help detect when the user has incorrectly typed in the badge number. This is automatically configured for every show and does not require any development besides alerting the user that they have mistyped the badge number whenever the Lead Response returns `{"error_code":113,"error_message":"Checksum Validation Failed"}`.

Please note that when submitting a Manual Lead Entry, the QR Validation does need to be

included as part of the request.

## Lead Request Submission (Live)

All queries will be POSTed to:

```
https://www.compusystems.com/servlet/LeadAPIRequest?type=submit
```

A JSON-formatted request to be POSTed to the API url will take the format the example shown below:

### Scanned QR Code Request

```
{
  "evt_uid":"8760",
  "badge_id":"5100356",
  "first":"M",
  "last":"C",
  "token":"b0883f925af870786df58ffc4105d107"
}
```

### Manual Entry Request

```
{
  "evt_uid":"8760",
  "badge_id":"51003564",
  "manual":"Y",
  "token":"b0883f925af870786df58ffc4105d107"
}
```

| item | Description |
|---|---|
| **evt_uid** *(event id)* | The unique identifier for the event for which you purchased the Lead API service. You'll receive this ID along with your token. |
| **badge_id** | A 6 to 7 character string retrieved from each badge that is scanned. |
| **first** | The first character of the registrant's last name. Applies only to QR code scan. You may submit first or last or both, but at least 1 must be included. |
| **last** | The first character of the registrant's last name. Applies only to QR code scan. You may submit first or last or both, but at least 1 must be included. |
| **token** | The token you received when you purchased the Lead API service. |
| **manual** | 'Y' indicates the request is a "Manual entry" request |

## Lead Response

Shown below is an example of the data you'll receive from a given badge:


```
{"First Name Prefix":"","First Name":"Meredith","Middle Initial":"","Last
Name":"Courtemanche","Last Name Suffix":"","Title":"Assistant
Editor","Company":"Pennwell","Company 2":"","Address":"98 Spit Brook Road","Address
2":"","Email Address":"test@test.com","City":"Nashua","State":"NH","Mail
Code":"03062-5737","Country Code":"","Country":"","Phone Area Code":"603","Phone
Number":"5555555","Fax Area Code":"630","Fax Number":"4214214","Badge
Number":"5100356"}
```


## Token

The token is valid only during a certain time period:  90 days before the show floor opens and 24 hours after the show floor closes.

## QR Validation

To ensure that the Lead Request is valid and to prevent Phishing, the requested badge number and the first letter of the first name and/or the first letter of the last name from the QR code is compared to the show database.  If they match you will receive the Lead Response; if they do not match you will receive an error response.

## Multiple Failures/Suspicious Activity

An unusually large number of failures or other suspicious activity can result in the suspension of your token.  This is done to help guard against "phishing" or other unwanted activity.

## Errors

Lead API uses the following error responses to communicate with users.

*Error Responses:*

99 = Missing argument 'type' expected
100 = Invalid argument 'type' specified
101 = Expected JSON in the HTTP POST body or a 'req' request parameter
102 = Missing argument 'evt_uid' expected
103 = Argument 'evt_uid' is a numeric only field
104 = Missing argument 'badge_id' expected
105 = Argument 'badge_id' is a numeric only field
106 = Missing argument 'token' expected
107 = Token Suspended

108 = Show Expired
109 = Invalid Badge Id
110 = Failed Validation
111 = Lead Request Error (Contact CSI Support)
112 = Missing arguments 'first' and/or 'last' expected
113 = Checksum Validation Failed
999 = Unexpected Error (Contact CSI Support)

*Example:*

```
{ "error_code":107, "error_message":"Invalid Token" }
```

## Code Snippets

### PHP Example

```
<?php
$URL = "https://www.compusystems.com/servlet/LeadAPIRequest?type=submit";
$data =
'{"evt_uid":"8760","badge_id":"5100356","last":"C","token":"b0883f925af870786df58ffc4
105d107"}';
$ch = curl_init($URL);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
#curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
#curl_setopt ($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLINFO_HEADER_OUT, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json',
'Content-Length: '.strlen($data)));
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>
```

Visual Basic .NET Example

```
Imports System
Imports System.IO
Imports System.Net
Imports System.Text

Module Module1
    Sub Main()
        Dim response As String
‘ You may use first, last, or both initials in your request
        Dim data As Byte() =
Encoding.UTF8.GetBytes("{""evt_uid"":""8760"",""badge_id"":""5100356"",""first"":
""M"",""last"":""C"",""token"":""b0883f925af870786df58ffc4105d107""}")
        Dim request As WebRequest =
WebRequest.Create("https://www.compusystems.com/servlet/LeadAPIRequest?type=submit")
        request.ContentLength = data.Length
        request.ContentType = "application/json"
        request.Method = "POST"
        Using requestStream = request.GetRequestStream
             requestStream.Write(data, 0, data.Length)
             requestStream.Close()
             Using responseStream = request.GetResponse.GetResponseStream
                Using reader As New StreamReader(responseStream)
                    response = reader.ReadToEnd()
                End Using
             End Using
        End Using
        Console.WriteLine("Response is: " + response)
    End Sub
End Module
```

C# .NET Example

```
using System;
using System.Text;
using System.IO;
using System.Net;

namespace leadAPISnippet1
{
   class Program
   {
      static void Main(string[] args)
      {
         byte[] data =
Encoding.UTF8.GetBytes("{\"evt_uid\":\"8760\",\"badge_id\":\"5100356\",\"first\":\"M\",\"last\
":\"C\",\"token\":\"b0883f925af870786df58ffc4105d107\"}");
         WebRequest request =
WebRequest.Create("https://www.compusystems.com/servlet/LeadAPIRequest?type=submi
t");
         request.ContentType = "application/json";
         request.ContentLength = data.Length;
         request.Method = "POST";
         Stream requestStream = request.GetRequestStream();
         requestStream.Write(data, 0, data.Length);
         requestStream.Close();
         WebResponse response = request.GetResponse();
         Stream dataStream = response.GetResponseStream();
         StreamReader reader = new StreamReader(dataStream);
         String responseFromServer = reader.ReadToEnd();
         Console.WriteLine("Response is: " + responseFromServer);
         reader.Close();
         response.Close();
      }
   }
}
```

# Testing

Now that you know the basics, let's walk through how to test your token for a sample show.

***Note:*** While the details of what you see here compared to what you see for your own show will vary somewhat, the principles involved are identical.

## Preliminaries (Test Environment)

**Procedure:**

1. Enter the following curl command for a given badge scanned at your event:
   ```
   curl -X POST -H "Content-Type:application/json" -d '{"evt_uid":"<your event id>", "badge_id":"<registrant badge id>", "first":"<registrant first initial>","last":"<registrant last initial>","token":"<your token>"} '
   https://www.compusystems.com/servlet/LeadAPIRequest?type=submit
   ```

   *Example:*

   ```
   curl -X POST -H "Content-Type:application/json" -d
   '{"evt_uid":"8760","badge_id":"5100356","first":"M","last":"C","token":"b0883f925af870786df58ffc4105d107"}'
   https://www.compusystems.com/servlet/LeadAPIRequest?type=submit
   ```

2. If your token is working correctly, you'll receive information from the scanned badge. *(See the sample info shown below.)*

   ```
   {"First Name Prefix":"","First Name":"Meredith","Middle Initial":"","Last Name":"Courtemanche","Last Name Suffix":"","Title":"Assistant Editor","Company":"Pennwell","Company 2":"","Address":"98 Spit Brook Road","Address 2":"","Email Address":"test@test.com","City":"Nashua","State":"NH","Mail Code":"03062-5737","Country Code":"","Country":"","Phone Area Code":"603","Phone Number":"5555555","Fax Area Code":"630","Fax Number":"4214214","Badge Number":"5100356"}
   ```

Test Information

**Evt_Uid *(Event id)***

8760

**Token Info**

*Token:* b0883f925af870786df58ffc4105d107

**Lead Submission URL for Testing Purposes**

https://www.compusystems.com/servlet/LeadAPIRequest?type=submit

**Lead Submission URL for Live**

https://www.compusystems.com/servlet/LeadAPIRequest?type=submit

## Sample QR Codes (Badges)

***Note:*** *The sample QR codes shown on the following pages are designed to work with the test token info only. (These will be tested to ensure they are working correctly.) Badges from onsite should be used for testing live.*

**QR Code 1 (Printed Badge)**

CSICSI|5100356|Meredith|Courtemanche

***POST Example for QR Code 1***

```
{
  "evt_uid":"8760",
  "badge_id":"5100356",
  "first":"M",
  "token":"b0883f925af870786df58ffc4105d107"
}
```

## QR Code 1 (Digital Badge)

CSICSI|5100356|Meredith|Courtemanche||5100356:V:345676812

### *POST Example for QR Code 1*


```
{
  "evt_uid":"8760",
  "badge_id":"5100356",
  "first":"M",
  "token":"b0883f925af870786df58ffc4105d107"
}
```


## QR Code 2 (Printed Badge)

CSICSI|155023|Susan|Mucha|

## QR Code 2 (Digital Badge)

CSICSI|155023|Susan|Mucha||155023:V:12478997201

## QR Code 3 (Printed Badge)

CSICSI | 238969 | Lea | Ban | 00000322 |

## QR Code 3 (Digital Badge)

CSICSI | 238969 | Lea | Ban | 00000322 | 238969:V:1619

## QR Code 4 (Printed Badge)

CSI689|155918|David|Hillman|

## QR Code 4 (Digital Badge)

CSI689|155918|David|Hillman||155918:V:45689236

## QR Code 5 (Printed Badge)

CSI87932 | 239033 | Sarajo y| Pickholtz | 00000285

## QR Code 5 (Digital Badge)

CSI87932 | 239033 | Sarajo y| Pickholtz | 00000285 | 239033:V:426

**QR Code 6 (Printed Badge)**

`csicsi|154713|Kazuhiko|Niwano|`

**QR Code 6 (Digital Badge)**

`csicsi|154713|Kazuhiko|Niwano||154713:V:1619472985170`

## Sample Manual Entry Badge Numbers (`badge_id`)

***Note:*** *The sample Manual Entry  shown on the following pages are designed to work with the test token info only. (These will be tested to ensure they are working correctly.) Badges from onsite should be used for testing live.*

***POST Example for Badge Number 1***

```
{
  "evt_uid":"8760",
  "badge_id":"51003564",
  "manual":"Y",
  "token":"b0883f925af870786df58ffc4105d107"
}
```

**Badge Number #1**
51003564

**Badge Number #2**
1550234

**Badge Number #3**
2389692

**Badge Number #4**
1559185

**Badge Number #5**
2390338

**Badge Number #6 (Failed Checksum Validation)**
154713

***Note:*** If you have any questions about QR Codes - or if you require additional samples - please contact Customer Service at **exhibitor-support@csireg.com**